

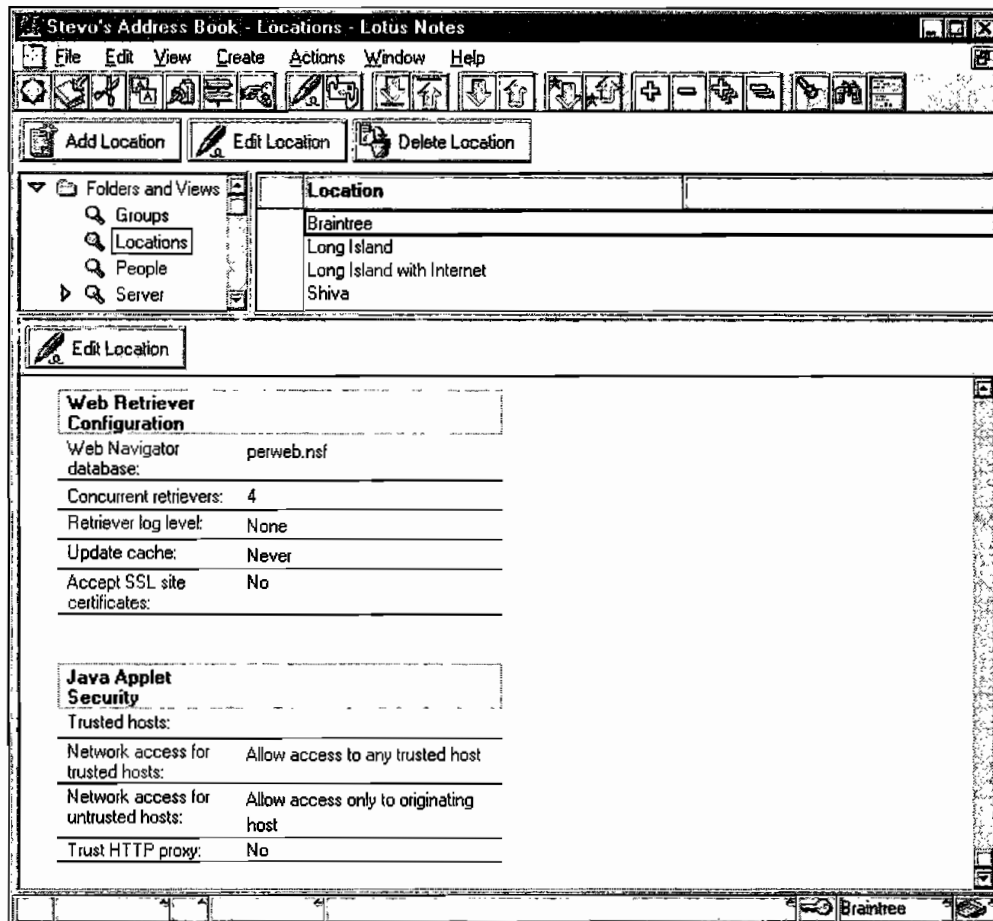


**Figure 16.4** Some of the configuration options are stored in Advanced section of the workstation's Location document.

The Web Navigator database field is used to determine two things. First, this field's contents determine the file name that is used when the Web Navigator database is created by the workstation. Second, the Notes workstation saves Web pages that the user has opened into this database. It stands to reason that the Web Navigator will not be able to save Web pages for the user if this database is missing, or if the name of the database is not listed properly in this field.

The Concurrent retrievers field tells the workstation how many of the operating system's processing threads can be dedicated to retrieving Web pages. The nature of the Web requires that a separate HTTP request must be sent from the workstation to the Web server for each element of a page; this means that a Web page that has, say, fifteen

images, would require several (at least fifteen) separate HTTP requests. The higher the number entered in this field, the more quickly the workstation will be able to retrieve and display such complex Web pages. Before rushing off to configure this field to its maximum value of six, however, consider that the more processing threads given over to retrieving Web pages, the more network traffic produced, and the less processing power the workstation has available for its other work. In most cases, the default value of 4 is sufficient for this field.

Use the `Retriever log level` to determine how much information is entered into the workstation's log as pages are retrieved. The options for this field are **None**, **Terse**, and **Verbose**. Since many workstation users are completely unaware that they have a Log database in the first place, **Terse** (or even **None**) is usually acceptable for the log level. Use the **Verbose** setting when trying to diagnose or troubleshoot a problem.

The `Update cache` field determines how often (if at all) the workstation checks the Personal Web Navigator database's version of a document against the version that's on the originating Web server.

As we have seen, when a user requests a Web page, the workstation checks to see if the document has already been retrieved and stored in the Personal Web Navigator database. If the document is not in the database already, the workstation retrieves the page from the Web server. If the page is already in the database, the workstation has to determine if it should go to the trouble of re-retrieving the page from the original Web server; the page will be re-retrieved only if the version of the document that's in the database is older than the version of the page that's on the original Web server. The default value for this field is `Once per session`; when a user requests a page that's already in the database, the workstation will compare its revision date against that of the original Web page on the original Web server, and only re-retrieve the document if the version that's in the Notes database is found to be out of date. Setting this field to `Every time` forces the workstation to always re-retrieve a page that's already in the database, and setting it to `Never` makes the workstation always "serve up" the version of a Web page that's in the database (without ever re-retrieving it).

Setting the `Accept SSL site certificates` to **Yes** means that the Notes workstation will be able to use SSL certificates as a way to authenticate Web servers. These certificates serve as the basis of secure Web

transactions, and represent a way for users to confidently transmit personal information such as telephone and credit card numbers. The use and configuration of SSL security is covered in Chapter 13, and in Appendix C.

Use the `Java Applet Security` section of the `Location` document to control how and where Java applets are run. Since there is no way of knowing exactly what an applet is going to do when run before it is downloaded by a user, there is significant security risk in these little programs. Use the `Trusted hosts` field to enter the name (or IP address) of Web servers; for example, you might use this field to enter *www.lotus.com*, assuming that any applets found on servers in that domain are trustworthy. In other words, you're betting that applets found on Lotus Development's Web servers won't do any harm to a user's workstation.

The `Network access for trusted hosts` field can be set to:

- Disable Java,
- Allow no access,
- Allow access only to the originating host,
- Allow access to any trusted host, or
- Allow access to any host.

The choice you make for this field determines how much access Java applets will have to the local computer. The safest option for this field is **No Access**, which will allow the applets to run, but will prevent such a program from accessing any of the information on the local workstation.

For example, consider the common and rather harmless applets that are used on many Web pages to display a scrolling marquee. Such an innocuous program does not need any access to the local system on which it is being run. On the other hand, applets such as those used on many stock ticker Web sites *would* require access to the local computer, since they must store (locally) the name of the stock that the user is tracking. (More detail about Java security is provided in Chapter 19.)

## THE WEB NAVIGATOR DATABASE

Cruising the Web (or opening Intranet-based pages) with a Notes workstation involves both the Search bar, and the database itself. Figure 16.5 shows how the screen looks for the typical Notes “surfer.”

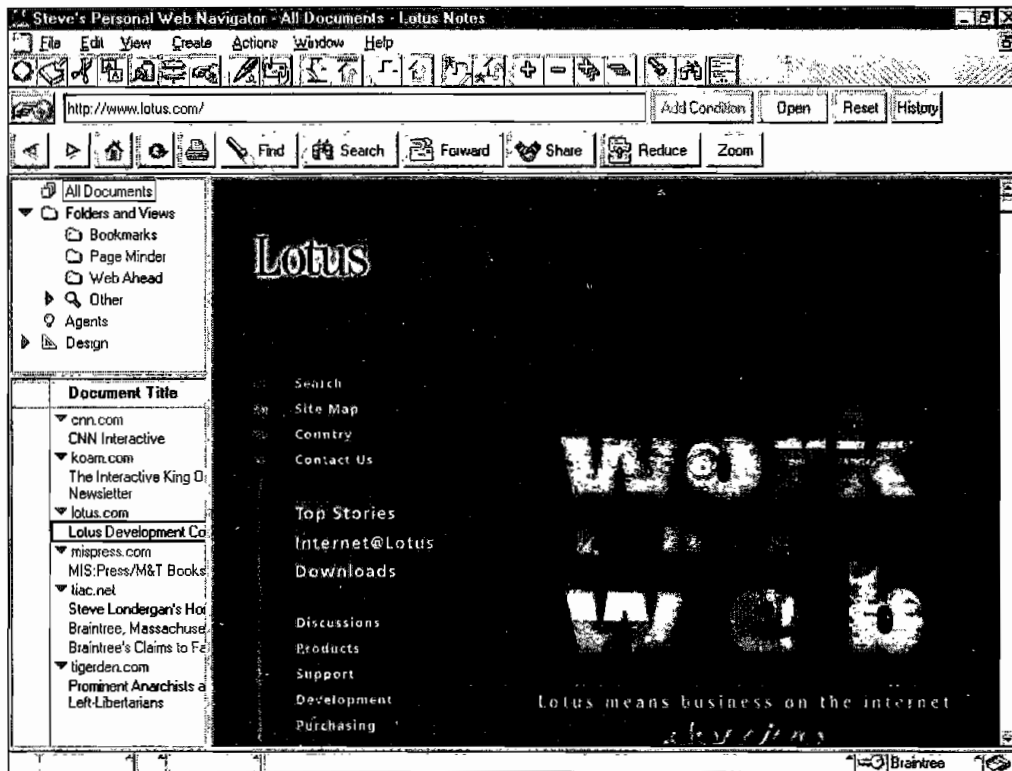


**Figure 16.5** Using Notes to surf the Web.

As illustrated in Figure 16.5, there are four parts of the screen that the user uses to interact with Web pages. The menus, in particular the Actions menu, hold several commands that are used to open and navigate through Web pages. Almost all of the functions available via the choices in the Actions menu are also available via the Action bar's buttons, so most users do not use the Notes menu whilst surfing.



## The Search Bar

When being used to surf the Web, the Search bar has a field for a user to manually enter the URL of a page; and it also displays the URL of the page that's currently open.

Since this same part of the screen is used both for entering a URL and for Full Text searches, this first button in the Search bar toggles back and forth between the Search bar used for performing Full Text searches, and that is used for URLs.

When a user has entered a URL in the URL field of the Search bar, he (or she) can press **Enter** to begin opening the page, or she (or he) can click the **Open** button. The **Reset** button will clear the contents of the URL field. The **History** button displays a list of the pages that have been already opened during the current session, and is an easy way to return to a page that was retrieved during the current session. Figure 16.6 shows the History dialog box.

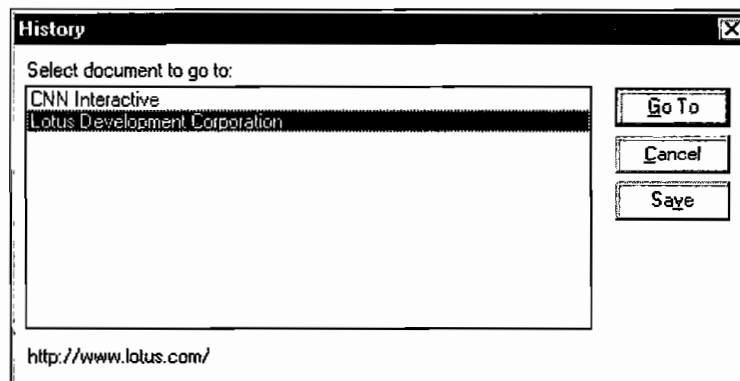


**Figure 16.6** The History button produces this dialog box, which can be used both to return to a previously opened page, and also to create a quick Web tour.

A Web tour is an easy way for the user to record the Web pages she has visited during a given session—and more important, to record them in the order in which they were visited. More details about creating and using Web tours is provided later in this chapter. Since all of the Web pages that a user opens are automatically stored in the user's Personal Navigator database, a Web tour is typically used only if the order in which they were visited needs to be preserved for some reason.

## The Action Bar

The Action bar in the Web Navigator database has eleven special buttons that are pre-programmed to perform certain Web-related activities. Refer to Figure 16.7.

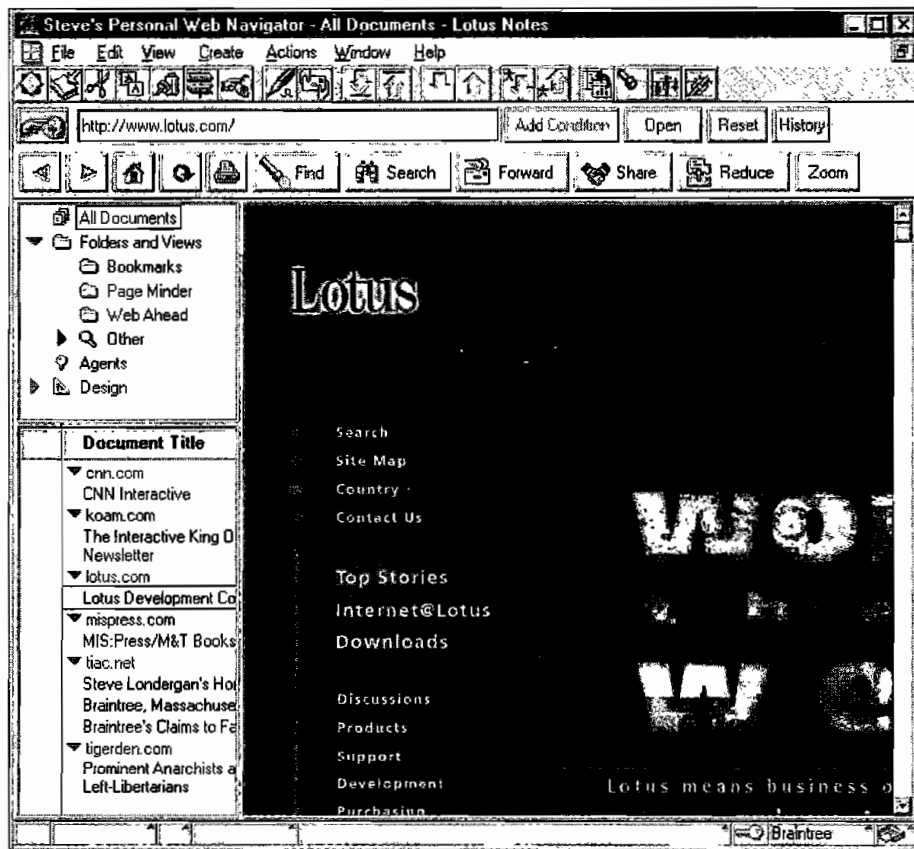


**Figure 16.7** Use the buttons in the Action bar to cruise the Web.

The Previous and Next buttons are used to return to the previous page that the user has opened or to go to the next page that has been opened. Since these buttons apply only to the pages that have been opened during the current session, they are of no use when the database is first opened. In other words, if the user comes into work in the morning, starts Notes, opens the Web Navigator database, and clicks the **Previous** button, they're not going to get anywhere!

The Home button opens the Web page that is listed as the user's Home Page in the Home page field in the database's Internet Options document. If your users are configured to surf a set of Intranet applications, this URL typically would be that site's internal master Web page; it could instead include your company's Home Page on the Internet, or any other page at all.



Remember that the Home page field in the Internet Options document for the database must include a URL for the Home button to work properly.

The Refresh button re-retrieves the page that is already opened and displayed on the workstation. When a user clicks this button, the workstation re-submits the HTTP request to the original Web server, and then re-retrieves the HTML page and re-converts it to a Notes document.

The Print button prints the current Web page.

The Find button is used to search for a text string on the current page.

The Forward button delivers the currently selected page to someone "else" an e-mail message.

The Search button opens the Web page for the user's preferred Internet search engine (such as LYCOS, Yahoo, Alta Vista, and so on) as specified in the Internet Options document for the database. If a page is currently open, and its HTML source code has been indexed, this button provides a quick way to search the page.

The Share button is used to, in effect, "publish" a Web page to the Public Web Navigator on an Domino server. The idea is that a user who has found a Web page that he considers to be of interest to other people can make that page available to them by copying the page to the server's copy of the Public Web Navigator database. Clicking this button produces the Share dialog box, as is displayed in Figure 16.8. The dialog box allows the user to forward the entire page, just the page's URL, and/or to assign a rating and category for the page.

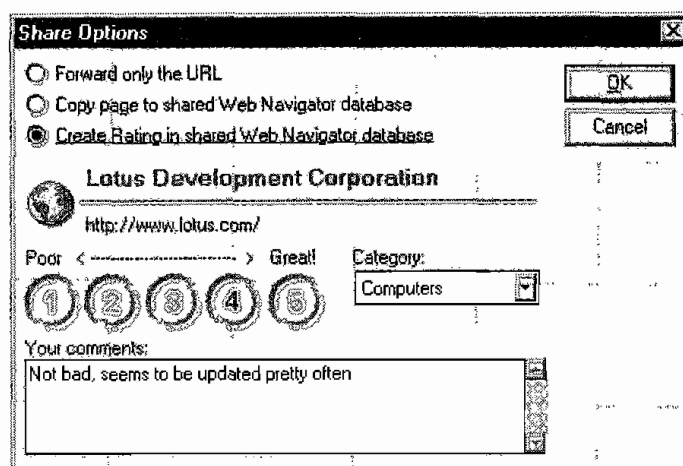


**Figure 16.8** Use the Share button to copy and rate a Web page from a workstation's Personal Web Navigator database to a server's Public Web Navigator database.



For the Share button to work properly, the fields in the *Collaboration* options section of the Internet Options document must include the name of the server and database that's to receive shared Web pages.

The Reduce button removes the Web page's text, links, and images—and preserves only a link to the page itself in the Personal Web Navigator database. This button is used to eliminate the page from the database (and recover the disk space it was using), while at the same time preserving the URL of the page for later use. Clicking this button forces the current page to be reduced, in much the same manner as the Database Purge Options that are set in the Internet Options document for the database.

The Zoom button simply maximizes the window for the Web page that is currently selected.

And, least but not least, the Folder button is used to put the current page into a folder in the database.

## The Views and Folders

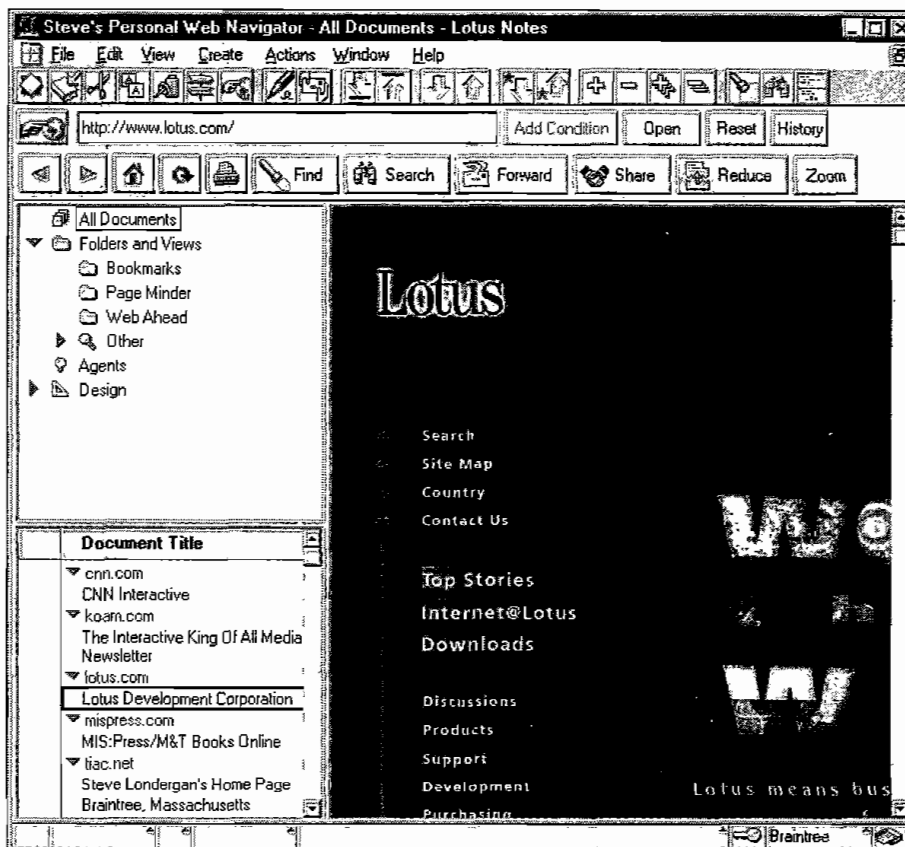


**Figure 16.9** The Views and Folders in the Personal Web Navigator database.

As with any other Notes database, the seven Views and Folders that are in the Personal Web Navigator database are used to sort, summarize, and organize the documents that are contained therein. Specifically:

- All documents lists (no surprises here) all of the Web pages that have been retrieved and stored in the database. The documents in this View are listed in order of host name.
- Bookmarks is a Folder that the user can employ to organize certain Web pages; the idea is that he (or she) will use the **Actions+Move to Folder** operations to put documents into this folder, and that they'll be easier to find later as a result.

- The **Page Minder Folder** contains the Web pages that are routinely and automatically checked for updates by the Page Minder agent. Any Web page that is moved into this Folder will be checked automatically whenever the Page Minder agent runs. More information about configuring and using this agent is provided later in this chapter.
- The **Web Ahead Folder** holds the Web pages that are automatically retrieved by the Web Ahead agent. Just like the Page Minder Folder, any Web page that is moved into this Folder will automatically be retrieved whenever the Web Ahead agent runs. More information about using the Web Ahead agent to automatically retrieve pages is provided later in this chapter.
- The **File Archive Folder** list documents containing any files that have been retrieved via FTP and stored in the database.
- The **House Cleaning View** is useful when the database has grown too large and the user is bent on manually deleting some documents to make the database smaller. Since this View lists the documents in order of their size, this is the best place to determine the largest documents in the database—and those which offer the most potential disk space to be recovered if deleted.
- The **Web tours View** lists any Web Tours (as described in Chapter 17) that have been saved in the database.

As with any other Notes database, users add new folders to their Personal Web Navigator database by using the **Create+Folder** operations. Documents can be moved from one Folder to another within the database by choosing **Actions+Move to Folder**.

## THE PAGE MINDER AGENT

The built-in Page Minder agent is a way for a Notes user to automatically monitor Web pages for changes, and to receive e-mail notification when a particular page has been changed. This agent is a great way to keep up with Web pages that are used to announce new software, new services, or even as a way to receive automatic e-mail delivery of Web pages from news services such as the New York Times

([www.nytimes.com](http://www.nytimes.com)), CNN ([www.cnn.com](http://www.cnn.com)), or perhaps any of the various on-line magazines via Pathfinder ([www.pathfinder.com](http://www.pathfinder.com)).

## Configuring the Page Minder Agent

All of the configuration options for the agent are listed in the Page Minder agent preferences section of the Internet Options document in the Personal Web Navigator database. (Open the database and choose **Actions+Internet Options** to edit it.)

As is displayed in Figure 16.10, the agent can be set to **Search for updates** once every hour, every four hours, once every day, or once every week. Obviously, checking for updates every hour produces a lot more network traffic and processing overhead than searching once a day. The default value for the field is every day; change this to one of the more frequent options only if the timeliness and “up to the minute” nature of the site being searched requires it.

Let's assume that the agent has been set to search for updates to the documents the Page Minder Folder once every day. When the agent searches for and finds a page that has changed since the last time the agent was run, it automatically re-retrieves the page and stores it in the database. The agent also automatically delivers an e-mail notification to the person whose name is listed in the **Send to** field. (Since the database is on the user's workstation, this field should contain the name of the person who own/uses the workstation, but could conceivably contain the name of a group of people who should be notified when the page in question has changed.)

The **When updates are found** field determines what type of notification is sent to the aforementioned user. Having the actual page mailed to the user means that he (or she) will find the whole Web page in their mail database; sending a summary instead mails the person a memo that includes a DocLink to the new and improved Web page that's been stored in the Navigator database's Page Minder folder.

Configuring the agent with the fields in the Page Minder agent preferences section of the document is only one-half of the story; use the **Enable Page Minder** button to turn on the agent, and then to determine where the agent will be run. Since the Personal Web Navigator database is, by definition, stored on the user's workstation,

this agent should always be run locally. For this (and any other) agent to run on a Notes workstation, **Enable local scheduled Agents** must be checked in the **File+Tools+User Preferences** dialog box.

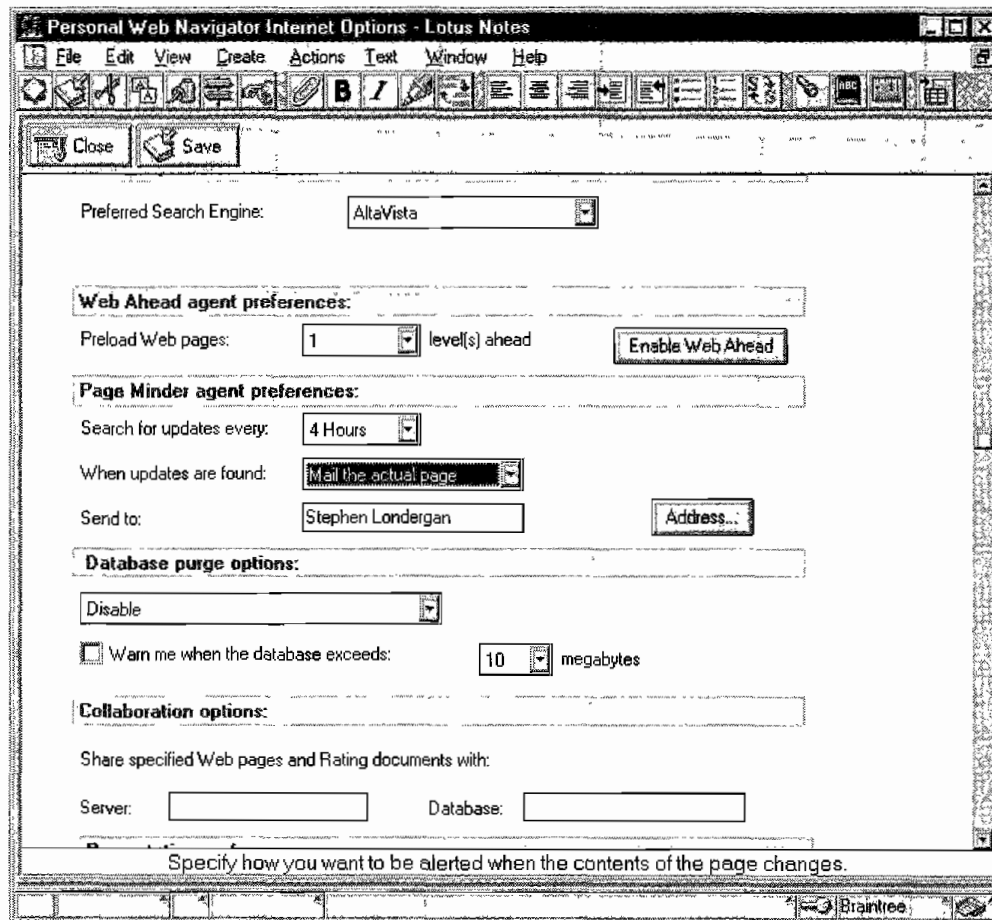


**Figure 16.10** Configuring the Page Minder agent to automatically search for and download Web pages that have changed.



**TIP**

Although the **Search for updates every** field is used to determine the frequency with which the agent checks Web pages for updates, the exact time when the agent is run can be changed by editing the agent itself (as you would edit or change any other agent in any other Notes database.) Open the database, choose **View+Agents**, double-click on **Page Minder**, and then use the **Schedule** button in the Agent Design dialog box to select the times when the agent will be run on the workstation.



## Choosing which Documents are Checked for Changes

Selecting a Web page to be routinely checked by the Page Minder agent is simply a matter of dragging the document in question into the Page Minder Folder (or choosing **Actions+Move to Folder** to get it there). Whenever the Page Minder kicks off and begins looking for pages that have changed, it uses the pages in the Page Finder as the basis of the search. Web pages that are found to have changed since the last time the agent was run are automatically retrieved and stored in the database, and an e-mail notification as defined in the Internet Options document is then sent. Remove the page from the Page Minder folder to stop the agent from checking it each time the agent is run. Figure 16.11 shows a sample e-mail notification about a changed page. (This one includes just a link to the document that has changed, rather than the entire document.)

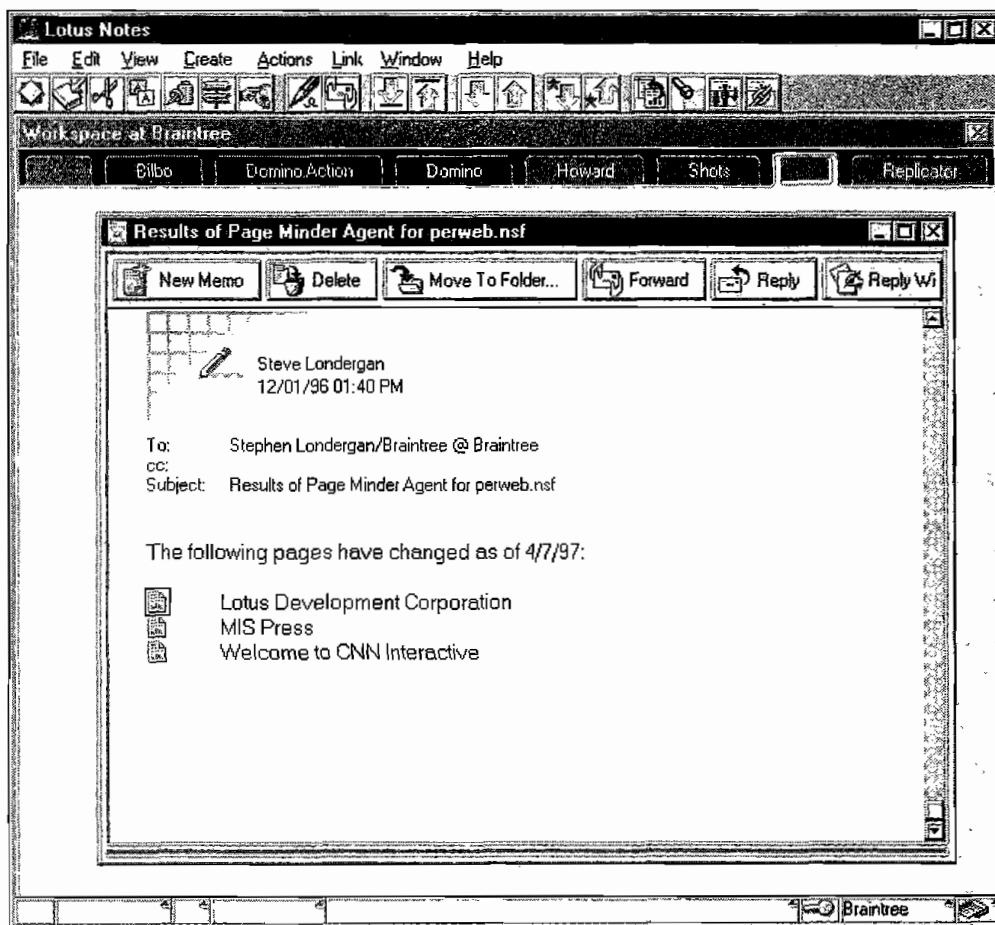


**Figure 16.11** The Page Minder agent sends an e-mail notification automatically when it finds a page that has changed since the last time the agent was run.

## USING WEB AHEAD

In addition to the Page Minder agent, the Personal Web Navigator database comes with another built-in agent, which is called Web Ahead. Similarly to the Page Minder agent, this agent uses a special folder in the database to determine which pages it should retrieve. When a document is placed into the Web Ahead folder, this agent automatically retrieves all of the pages to which the given page has Links. And, depending on how the Web Ahead agent has been configured, it will retrieve and store the pages to which those pages have Links, too.

The idea behind this agent is to automatically retrieve and store as many pages as possible for off-line use—imagine a laptop computer that is connected to the Internet during the day, and which runs the Web Ahead agent. At the end of the day, when the owner takes the laptop home on the train, she (or he) can open the database and read the Web pages that have been stored in it; in most cases, the Links on the pages will even work, since the Web Ahead agent automatically retrieves the linked pages.

All of this makes for an excellent way to read pages and catch up on the WWW even while a user is disconnected, but it also provides the fastest and most direct way to fill up a hard disk! It's impossible to predict how much space will be taken up by the pages that are automatically retrieved by the Web Ahead agent. Imagine using the Web Ahead agent to retrieve the pages on Microsoft's Web site. Since many of the pages on that site have embedded files, running the Web Ahead agent there could have disastrous results, as it merrily goes about its business of downloading sixty-five copies of Windows, driver updates, new software from the publisher, and so on.

### Configuring the Web Ahead Agent

Just like the Page Minder agent, the Web Ahead agent is configured via the Internet Options document in the Personal Web Navigator database. As is displayed in Figure 16.12, there's only one field used to configure this agent: Preload Web pages. This field (which can be set to 1, 2, 3, or 4 levels ahead) determines how many levels "deep" the agent will go when retrieving pages.

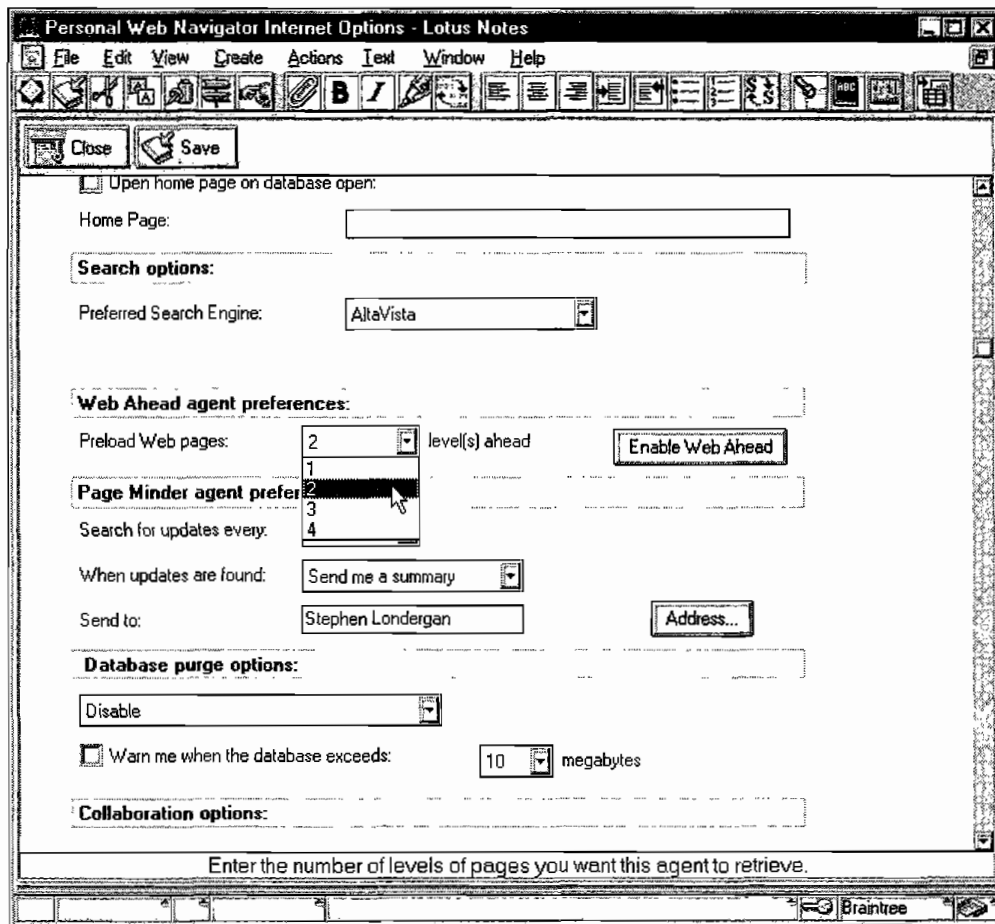


**Figure 16.12** The Internet Options document in the Personal Web Navigator database is used to configure the Web Ahead agent.

For example, let's say that the Web Ahead agent was configured to preload four levels of Linked pages, and that the Web Ahead Folder contained the Home Page for MIS Press (*www.mispress.com*). Such a scenario would produce a Web Navigator database that contained hundreds of Web pages, and that took up many megabytes of disk space. Since it is virtually impossible to predict "where" the links on a Web page are going to "go," (to say nothing of how many links will be found on a given page), use caution when setting this field to anything more than the default value of 2. Many the person has been surprised to find *an entire Web site* stored in the Personal Web Navigator database after setting the Preload field too high!

Configuring the agent with the fields in the Web Ahead section of the Internet Options document is only one-half of the story; it's also necessary to use the **Enable Web Ahead** button to turn on the agent, and also to determine where the agent will be run. Since the Personal Web Navigator database is, by definition, stored on a user's workstation, run this agent locally. For this (and any other) agent to run on a Notes workstation, **Enable local scheduled agents** must be checked on the **File+Tools+User Preferences** dialog box.

**T I P**

By default, this agent is scheduled to run every hour. Control the exact time when the agent is run by editing the agent itself: open the database, choose **View+Agents**, double-click on **Web Ahead**, and then use the **Schedule** button in the Agent Design dialog box to select the times when the agent will be run on the workstation.

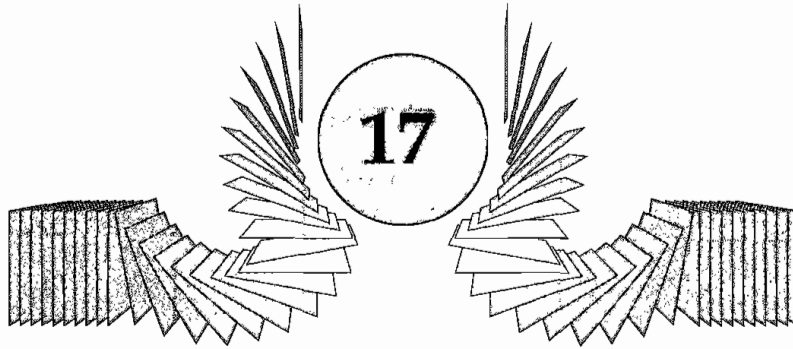
### Choosing which Documents are Automatically Retrieved

To select a Web page and the corresponding Linked pages which will be automatically retrieved and stored by the Web Ahead agent, drag the document in question into the Web Ahead Folder (or choose **Actions+Move to Folder**). Whenever the Web Ahead agent runs, it retrieves the pages in this folder. Remove the page from the Web Ahead Folder to prevent the page (and its corresponding Linked pages) from being retrieved in the future.

### SUMMARY

- When a workstation is configured to retrieve Web pages with the Notes workstation software, every Web page that a user retrieves is automatically stored in the Personal Web Navigator database, which provides an easy way to read the page "later" without having to re-retrieve the page from the Web.
- The built-in Page Minder agent can be used to automatically search for and retrieve Web pages that have changed; use this agent to "keep up" with Web pages that change frequently.

- The Web Ahead agent loads Web pages into the Personal Web Navigator database so that the pages can be viewed or read after the workstation is disconnected from the Internet (or Intranet).
- The Internet Options document in the Personal Web Navigator database is used to configure the manner in which Web pages are retrieved and the frequency with which the agents are run.
- The Internet Options document also can be used to help manage disk space by configuring the Purge options.



# **USING AND MANAGING THE PUBLIC WEB NAVIGATOR DATABASE**

When workstations are configured to use a Domino server to retrieve Web pages, the pages they retrieve are automatically stored in the Public Web Navigator database on the server. Like its diminutive cousin the Personal Web Navigator, this database also contains agents that are used to automatically manage the pages in the database, and a special "Refresh" agent that ensures the pages in the database are kept up-to-date.

Since each and every Web page ends up in this database, Web-related Internet network traffic is kept to an absolute minimum. Once a given page has been stored in the Public Web Navigator database, any subsequent request for that page, from any user, can be "served up" from the database, and does *not* require an Internet connection to the original Web server. And since all of the pages requested by users are stored in the database, the chances of any given page having already been retrieved and stored in the database go up as the database is used more and more; it's just a matter of time before a lot of the Web pages that users at a given location or company will want are already in the database.

In this chapter, you'll learn about using a Notes workstation to open Web pages via a Domino server, about using, configuring, and managing the Public Web Navigator database, and how to configure and use the Refresh agent.

## THE BIG PICTURE

When Notes users workstations are configured (via the Location documents in their Personal N&A Books) to retrieve Web pages through the Domino server, each page the users open is automatically stored in the Public Web Navigator database. The good news is that the Domino server remembers which pages have been previously opened and stored in the Navigator database, so that subsequent requests for the same page do not necessarily initiate an Internet (or Intranet) connection to the original Web server. Instead, the server finds the version of the page that's already in the Navigator database and displays *it* to the user. This results in a lot less LAN/WAN traffic as more and more Web pages can be "served up" from the database instead of from a Web server.

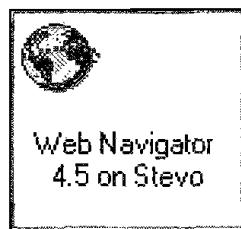


**Figure 17.1** The Public Web Navigator database on a Domino server is used to store and manage Web pages for multiple Notes users.

## About the Database

The database itself is similar to other Notes databases, and contains the same types of Folders, Views, Forms, and Action bar buttons that are already familiar to most Notes users.

When a Notes server is installed, the template that will be used as the basis of the server's Public Web Navigator database is automatically

copied into the server's data directory. The name of the template file is `pubweb.ntf`. When the Web retriever task is loaded on the Domino server for the first time, the server creates a Public Web Navigator database, the file name for which is (usually) `web.nsf`. (The exact file name for this database is prescribed by the Web Navigator database field in the Web Retriever Administration field in the Web Retriever Management section of the server's server document in the Public N&A Book.)

The default file name for a Public Web Navigator database is `web.nsf`, but this can be changed, if necessary. Modifying the contents of that Web Navigator database field in the aforementioned section of the Public N&A Book will "force" the server to create a Web Navigator database by that name the next time the **Web Retriever** task is started on the server.

As we have seen, when a user issues a request for a Web page, the Domino server retrieves the page by invoking a special server task, known as the *Web Retriever*. This task must be loaded on the server for users to be able to retrieve Web pages; Chapter 15 discusses changing a server's `NOTES.INI` to add this task to the `ServerTasks=` line and load the task automatically every time the server is started.

## Opening a Web Page

One of the central themes of Notes is that users have access to the Web any time, anywhere. Consequently, there are five ways that a workstation can open a Web page:

- By using the **Open URL SmartIcon**
- By choosing **File+Open URL** (or pressing **Ctrl+L**)
- By entering the URL and then clicking the **Open** button in the Search Bar
- By clicking a button or Hotspot that has been programmed to open a URL
- Via an agent

A user employing any of these five methods gets the Web page displayed on her workstation, and also (perhaps unknowingly) stores



the page in the Domino server's Public Web Navigator database. (Remember that if a user requests a page that's already been stored in the Public Web Navigator database, the page is retrieved from there instead of from the Web.)

When a Web page is being retrieved by the Domino server, a blank page with an animated icon of some notes of paper floating over a map is displayed on the workstation, as is shown in Figure 17.2.

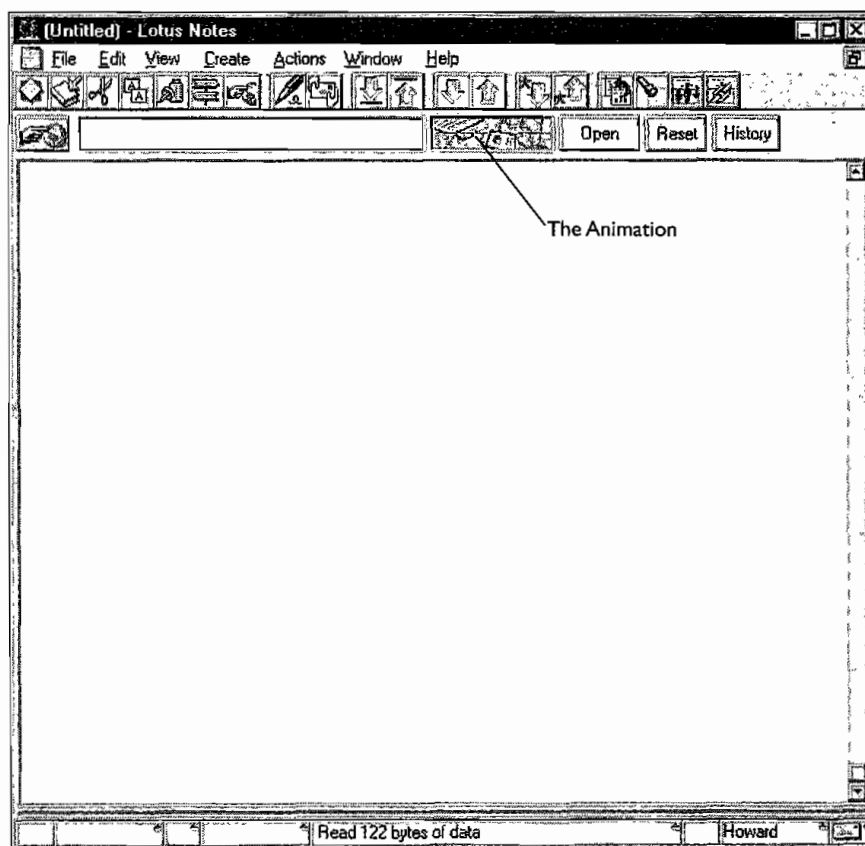


**Figure 17.2** The user sees these fluttering pages while a Web page is being retrieved by the Domino server.

Users can click on the animation to cancel the retrieval of, and stop waiting for, a page. And the **Window** menu command is available while a page is being retrieved, so the user can switch to other documents, databases, and even servers while waiting for the page to finish.

By default, the page is not displayed until it has been retrieved in its entirety; choose **File+Database+Properties** and click the **Display images after loading** box to configure the database to display images *after* retrieving and displaying the text for the page. This option is significant, as it gives users the impression that the page has loaded more quickly than it actually has.

## CONFIGURATION

The configuration options used to determine how Web pages are retrieved by a Domino server and displayed on a Notes workstation are stored in three places: the users' Location documents, the Server document for the Domino server in the Public N&A Book, and finally in the Internet Options document that's stored in the Personal Web Navigator database itself.

### Configuring and Administering Workstation Location Documents

As discussed in Chapter 15, a workstation is configured to use a Domino server for Web page retrieval via the Location documents in the Personal N&A Book. The Location document contains the fields that are used to define Notes as the Web browser, that pages should be retrieved from the Domino server, and to indicate the name of the Domino server from which the pages will be retrieved.

Create a *Setup profile* that can be used when registering a new user to automatically configure the new user's workstation for Web page retrieval via an Domino server. (Create the new Setup profile by clicking the **Add Setup Profile** button that's found in the Action bar for the Setup Profiles View in the Public N&A Book.) As is displayed in Figure 17.3, this form allows you to choose **Notes** as the **Internet browser** and **from Domino Server** in the *Retrieve/open pages* field.

**Setup Profile: LAN Users - Lotus Notes**

File Edit View Create Actions Text Window Help

Close Save

**USER SETUP PROFILE: LAN Users**

**Basics**

Profile name: LAN Users

Internet browser: Notes

Retrieve/open pages: from InterNotes server

**Default Databases**

Database links:

**Default Passthru Server**

Server name:

Country code:

Area code:

Phone number:

**Default Connections to Other Remote Servers**

| Server Names | Country Codes | Area Codes | Phone Number |
|--------------|---------------|------------|--------------|
|              |               |            |              |

**Default Secondary**

Enter the name for this profile. e.g. Sales

Brantree

**Figure 17.3** Use a Setup Profile to automatically define the browsing options for a new Notes user.

When registering a new user, use the Profile list on the Register Person dialog box to specify the aforementioned Setup Profile document from the Public N&A Book. When the user later installs his Notes workstation, the browsing options that are defined in this Setup Profile will be copied to the new user's Personal N&A Book, saving the user from having to manually edit his Location document—and also, perhaps, saving him from having to learn what a Location document is in the first place.

**TIP**

If a workstation has been configured to use an Domino server for Web page retrieval, and the Domino server field on the workstation's Location document has been left blank, the workstation will default to using the user's Home server as the Domino server.

The bad news is that there's no way to centrally manage Location document settings for existing users. For example, imagine a scenario that has, say, 300 users whose Notes workstations have already been configured to use their Home servers as the Domino server. For the sake of the argument, we'll say that there are three Notes servers supporting these three hundred users. As use of the Web Navigator has grown (not to mention the size of the three Web Navigator databases on each of the three servers), you've decided to centralize the Web users to a single, optimized Domino server for Web access. Such a move reduces the network and processing load on the three Home servers, reduces in size and consolidates the three versions of the Public Web Navigator database, and centralizes the Web-related administrative tasks to a single server.

At first blush, such an undertaking seems simple—after all, it's relatively easy to remove the Web Retriever task from each of the three server's NOTES.INI files, and installing the new Domino server isn't so difficult, either. The problem arises with the necessary reconfiguration of the workstation's Location documents to point to the new Domino server for Web access. Since there's no centralized way to “reach out and touch” each of the 300 Location documents to insert the name of the new Domino server, the worst-case scenario requires each user to manually edit his/her own Location document. Given the aptitude (or lack thereof) many users have for configuring their workstations, this is no mean feat.

One possible solution would be to send each of the users an e-mail message that included a button pre-programmed to open the user's Personal N&A Book, edit the appropriate Location document, and insert the new Domino server's name into the appropriate field. Even with this cute little e-mail button, it's plain to see that changing a workstation's configuration to use a different Domino server is no picnic.

## Configuring the Server Document

Typically, two parts of a Server document in the Public N&A Book are used to configure the Domino server: the `Web Retriever Administration` section, which defines such options as the name of the Web Navigator database, the number of threads used for Web page retrieval, and so on; and the `Proxy Configuration` section, which, optionally, is used to specify the URL or IP address of the Internet Proxy server that the Domino server should use for Internet connections. The use of the `Proxy Configuration` section of the Server document is covered in Chapter 24.

## WEB RETRIEVER ADMINISTRATION

Although the basic concepts of configuring a Domino server's Server document were discussed in Chapter 15, there are a few finer points that bear mention here. The first, and arguably most important, field in this section of the Server document (as displayed in Figure 17.4) holds the name of the database that the Domino server will use to store and manage Web pages. This database is created automatically when the Web Retriever task is loaded on the server for the first time, and the icon for this same database is added to a workstation's desktop the first time a Web page is opened on that workstation.

The Web Navigator database itself is used each and every time a user opens a Web page, whether the user realizes it or not. Even a simple click of the **Open URL** SmartIcon requires that the user doing the clicking have access to this database, since the workstation and the Domino server have to work together for each and every HTTP request

to determine whether or not the Web page being requested has already been stored in the Web Navigator database.

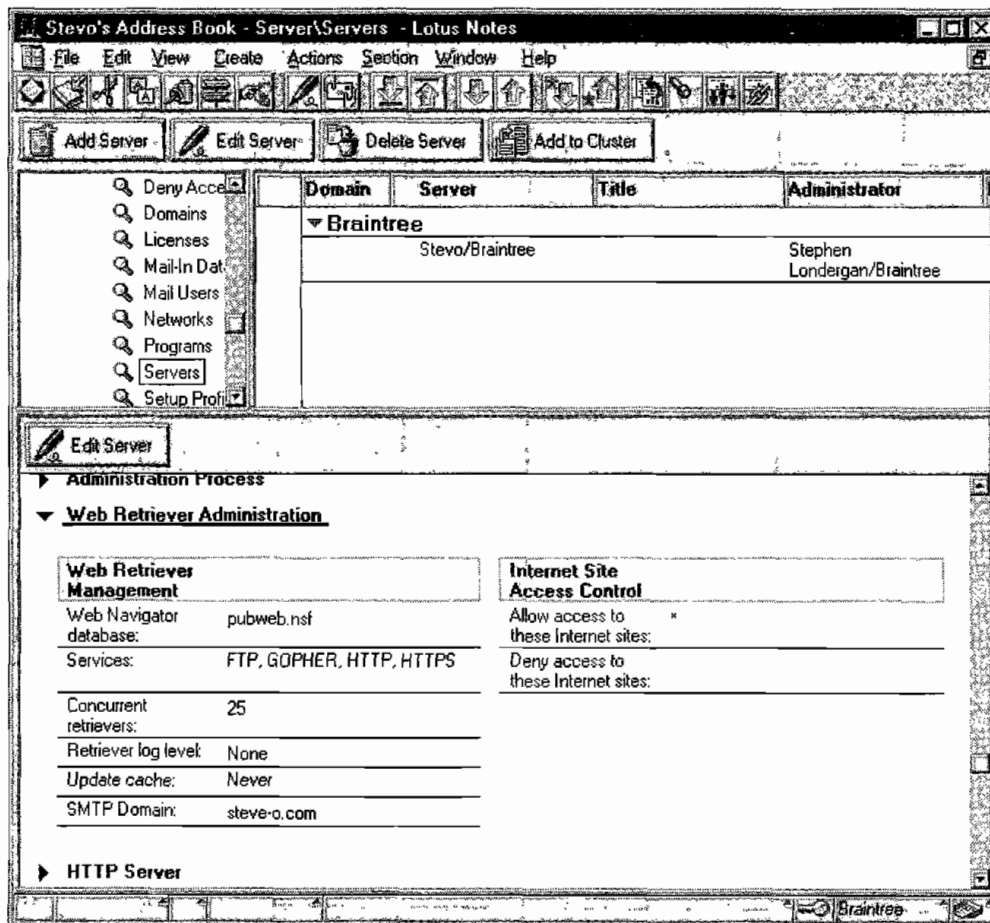


**Figure 17.4** The Web Retriever Administration section of a Server document in the Public N&A Book.

As is discussed later in this chapter, the database contains Views, Folders, and agents that make it an application in and of itself; the database can even be customized to include other special Forms, Views, and so on that make it particularly useful or relevant to a particular company's use of the Web. Some users could use the documents in the database without necessarily using the World Wide Web at all.

On the other hand, some users may never open this database, and may not need to use the views and folders contained therein.

In any event, the Access List for this database is as important as the Access list for any Notes application, and careful consideration and planning should go into the levels of access that are assigned to users. In most cases, the Default level of access to the Public Web Navigator database is set to **Editor**, which allows users to create and edit the Web pages they find in the database.

The *Services* field on the Server document determines what kind of URL requests can be submitted by workstation users to the Domino server. Administrators concerned about the potential virus threats inherent in allowing users to download files via FTP have been known to eliminate the FTP service from the Domino server; doing so prevents users from using the Domino server to download files from FTP sites. (Although a user could still "get" a virus by downloading a file from a Web site; this option disables just *FTP* requests.) See Figure 17.5.

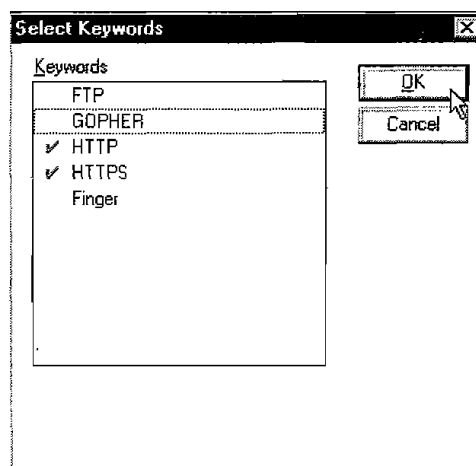


**Figure 17.5** Selecting which types of URL will work on a Domino server.

Of special interest in the `Service` field is the ability to enable **HTTPS** connections. As discussed in Chapter 13, a Web page address that is prefaced with **HTTPS:** implies that the Domino server requesting the page and the Web server that has the page will be able to exchange certificates to authenticate one another. Enabling the HTTPS service on an Domino server does not mean that all transactions will be authenticated in this manner; rather, it means that when a user retrieves a URL that uses the HTTPS convention, that session between the Domino server and Web server will use the SSL Authentication protocol.

Enabling the HTTPS service on an Domino server also requires the appropriate entries and selections in the `SSL Settings` part of the `Security` section of the Server document.

The `Concurrent retrievers` field determines how many of the operating system's processing threads on the Domino server are given over to Web page retrieval at any one time. It stands to reason that a Notes server that is used for nothing but Web page retrieval could probably withstand an increase in this field to maximize page retrieval performance. At the same time, using a single Notes server for applications, mail databases, *and* Web page retrieval might warrant a reduction of the value in this field to more evenly balance the processing power amongst the server's tasks.

The rest of the fields in the `Web Retriever Management` section of the server document are explained in full detail in Chapter 15.

## USING THE PUBLIC WEB NAVIGATOR DATABASE

The Public Web Navigator database as it ships from Lotus is configured to open a Navigator when the database is opened, as you can see in Figure 17.6.



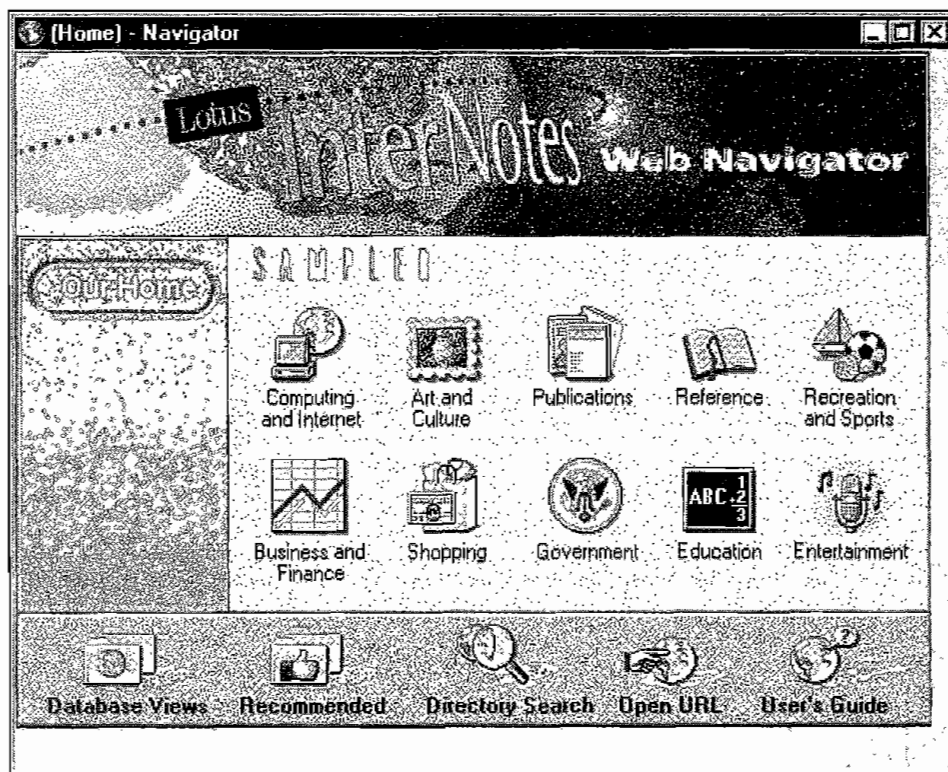


**Figure 17.6** You see this Navigator when you open the Public Web Navigator database.

### The Database's Main Navigator

This Navigator, and the buttons on it, are intended to be a sort of Home Page for users at your company or location.

The Home button in the upper left-hand corner of the Navigator is pre-programmed to open a Web page at IBM. (Chapter 18 lists the steps to customize this button to point to some other page on the Web, or perhaps to your organization's Intranet Home page.)

The various icons in the Sampler section of the Navigator are each links to folders bearing the same name within the database. The Web

Navigator database comes "pre-loaded" with the URLs for several pages in each of these categories; in other words, those fine folks at Lotus have already loaded the database up with some URLs for Web pages related to "Computing and the Internet," "Art and Culture," "Publications," and so on. Clicking any one of the buttons in the Sampler section of the Navigator opens the View in the database with the documents corresponding to that category. Users who choose to rate a page that they have found in the Web Navigator database also have the option of assigning any one of these categories to the rated page.

The first button at the bottom of the Navigator, Database Views, in turn opens another Navigator in the database (called the Views Navigator) that is used to navigate amongst the various Views and Folders in the database. The Recommended button opens a view of the Web pages that have been stored in the database and assigned a ranking by users.

The Directory Search button opens a special Form in the database that is used to perform Internet searches; this form is pre-programmed to use the Lycos, and Yahoo search engines.

The Open URL button produces the Open URL dialog box, which is the same one that is produced when the user chooses **File+Open URL** from the menu. This dialog box is used to manually enter the address of a Web page to be opened.

The **User's Guide** button opens the Help database for the Public Web Retriever database.

### **The Database's Other Views and Folders**

As discussed in the previous section, the Database Views button on the database's Home Navigator produces another Navigator, which is used to navigate and choose from the various Folders and Views in the database. Figure 17.7 shows this Navigator.

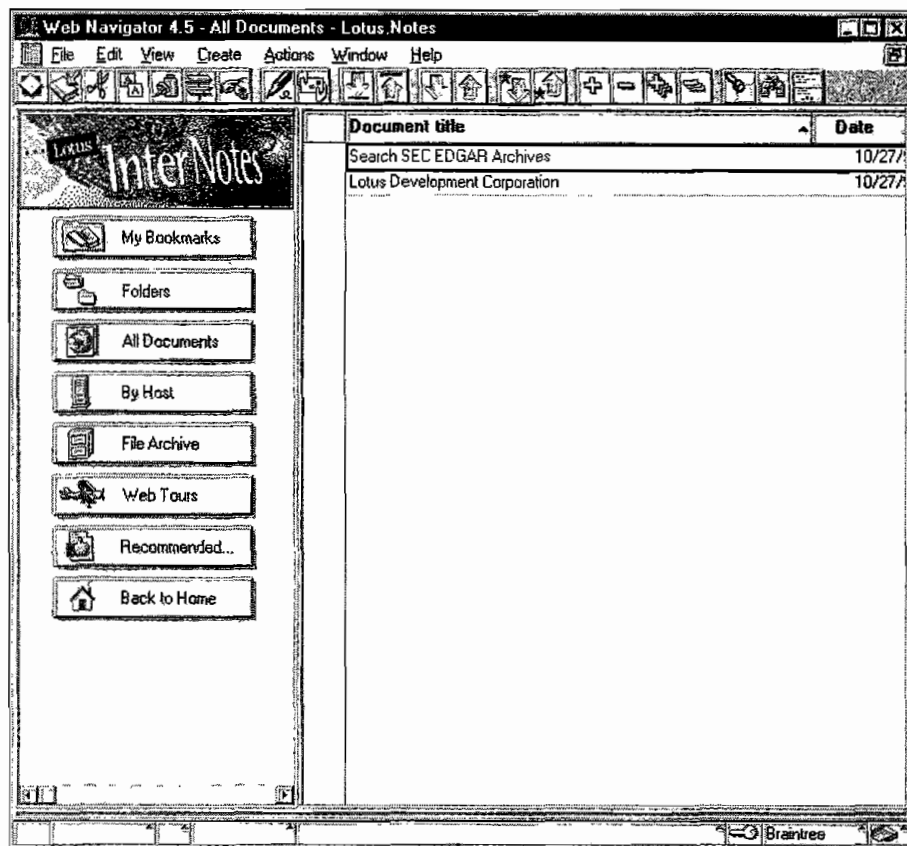


**Figure 17.7** The Views Navigator.

There are eight buttons on the Views Navigator:

- **My Bookmarks**, which opens a Private Folder for each user of the database. By default, the Web Navigator contains a Folder named My Bookmarks, which is designed to be made Private on First Use. In other words, each time a new user selects the **My Bookmarks** button on the Views Navigator, Notes creates a private folder named **My Bookmarks** for the user. The user can then use **Actions+Move to Folder** to put documents into the folder. This provides an easy way for a user to organize and collect Web pages from within a database. (Users can also create other private folders by choosing **Create+Folder**.)
- **Folders**, which simply opens the “normal” list of Views and Folders that are in the database.

- **All Documents**, used to list all of the Web pages that have been stored in the Web Navigator database, in alphabetical order based on the page's title.
- **By Host**, which lists the Web pages in the database sorted by the name of the Web server from which they were retrieved.
- **File Archive**, which lists the Web pages that have been stored in the database with file attachments. (Such file attachments are usually produced as the result of an FTP request.) Administrators looking to reduce the size of the Web navigator database by manually deleting documents should start with this View, as the documents that have attachments are obviously the largest and offer the most potential for disk space savings.
- **Web Tours**, which lists the documents created when users choose to save their Navigation histories.
- **Recommended**, which lists the Web pages that have been assigned a rating by users.
- **Back to Home**, which closes the Views Navigator, and returns to the database's Home Navigator.

## USING THE MANAGEMENT AGENTS

The Public Web Navigator has three agents that are used to help manage the database. The *Averaging agent* polls any documents in the database that have been rated by users to determine which should be included in the Top Ratings Folder. The *Purge agent* manages the size of the database, and the *Refresh agent* keeps the pages in the Web database up to date.

### Purging the Database

The fields in the Database purge options section of the Administration form are used to control the size of the Navigator database. It stands to reason that as more and more people use the Domino server to retrieve Web pages, the database will grow larger and larger. Consequently, the database has a built-in Purge agent that manages the size of the database. This agent is scheduled to run every night at 1:00 A.M.

Each night at 1:00 A.M. this agent decides whether to “work on” the database based on the `Maximum database size` field. If the agent finds that the database has grown to equal or exceed the value in this field, it will reduce the size of the database until it has reached the value in the `Purge to what % of maximum size` field. For example, if the maximum database size was set to 500 megabytes, and the `Purge to what % of maximum size` was 80%, the agent will keep on reducing or deleting pages until the database size is reduced to 400 megabytes. Once the agent determines that it has reached the appropriate database size, it stops reducing or deleting pages—until the next night at 1:00 A.M.

If, at 1:00 A.M., the agent determines that the database has not met or exceeded the maximum size, it does *not* proceed, and instead waits until the following day, when it evaluates the database all over again (at 1:00 A.M.).

The `Purge agent action` field determines what, if anything, the agent does when it determines that the database has grown too large. The options for this field are `Reduce page` and `Delete page`. Although the latter option is obvious, the former bears some explanation. If the `Purge agent` is using the `Reduce page` method, it removes the Web pages (along with their associated text, images, Java applets and so on) from the database, and replaces each with a URL link to the original Web page on the original Web server. (The `Delete page` method removes the page entirely, leaving no trace whatsoever.)

When the agent has determined that it needs to run, it first reduces or deletes the pages that have been stored in the database longer than the number specified in the `Purge documents older than` field. Very often, removing or reducing the old pages may be enough to reduce the database to the target size (Maximum database size multiplied by the `Purge to what %` field.) If, after removing the oldest pages from the database, the agent determines that the database is still too large, it then begins reducing or deleting the pages that exceed the `Purge documents larger than` field.

### Refreshing the Database

The *Refresh agent* is used to ensure that the Web pages in the database are up-to-date; since Web pages tend to change so frequently, using the

Refresh agent is almost a requirement to make sure that the version of a given page that's in the Public Web Navigator database is as current as the version of the page that is available on the originating Web server.

#### How the Refresh Agent Works

The agent guarantees "freshness" of Web pages by comparing the date when a given Web page was stored in the database with the date when the original Web page (on the originating Web server) was last changed. The Refresh agent automatically checks all of the pages in the database, with the exception of FTP and Gopher pages, and those that are stored only in the personal folders that have been created by users.

This agent follows a rather strict and orderly set of instructions for refreshing the pages:

- First the agent selects a page in the database to check.
- If the agent finds that the page in question was added to the database *today*—the day that the agent is being run—it does not proceed any further with the current page, and instead selects the next page in the database and begins to check it.
- If the page in question was added to the database before "today," the agent contacts the Web server on the Internet (or Intranet) from which the page was retrieved in the first place.
- The Refresh agent then compares the date when the page was added to the database with the modification date from the page's HTTP header on the Web server.
- If the Notes date is after the HTTP modification date, the agent does not proceed, and instead selects the next page in the database and begins checking it.
- If the HTTP modification date is found to be after the date when the page was added to the Navigator database, Notes assumes that the version of the Web page on the Web server is "new and improved." Consequently, the Refresh agent re-retrieves the Web page from the Web server, converts it into a Notes document, and replaces the outdated version of the page in the database with the new version.

The bottom line is that the agent does its best to ensure that the pages stored in the Public Web Navigator database are truly the latest and greatest. Also note that running this agent produces an appreciable amount of LAN/Internet traffic, since, at the very least, the agent must contact all of the Web servers to obtain the HTTP modification date for comparison with the Notes creation date. Even more traffic is produced when a number of pages are found to be out of date and must be re-retrieved by the agent.

#### Enabling the Agent

To turn the agent on, click the Action bar's **Enable Refresh agent** button when editing the database's Web Navigator Administration document. By default, this agent runs each day at 3:00 A.M.; change this time if necessary by using the **Schedule** button on the agent's Design pane.

### The Averaging Agent

The *Averaging agent* is used to sort through and organize the rated documents in the database. There are many Web sites in the world that produce "Top Pick" and "Best Sites" lists; think of this agent as a way to produce a similar, sorted-by-popularity, list of the sites that users have been visiting via the Domino server. The agent reviews the documents in the database that have been assigned a rating by the users to determine which pages have received the highest collective rating. In other words, this agent will not do anything if the users have not been rating the pages they've retrieved and read. Consequently, it may not be necessary to use this agent at all; if the majority users are *not* in the habit of using the Recommend button (or the Share button in the Personal Web Navigator), there's no reason to even run it. Domino servers have a lot of work to do, and if that load of work can be lessened by not using this agent...

#### How the Refresh Agent Works

By default, this agent is configured to run every hour. At the top of the hour, the agent reviews the documents in the database that have been rated, determines which of them is the most popular, and deposits the "Top Ten" Folder into the database.

**NOTE**

Pages that have not been assigned a rating by users, and the pages that have been automatically retrieved by other agents in the database, are not factored into the Most Popular list.

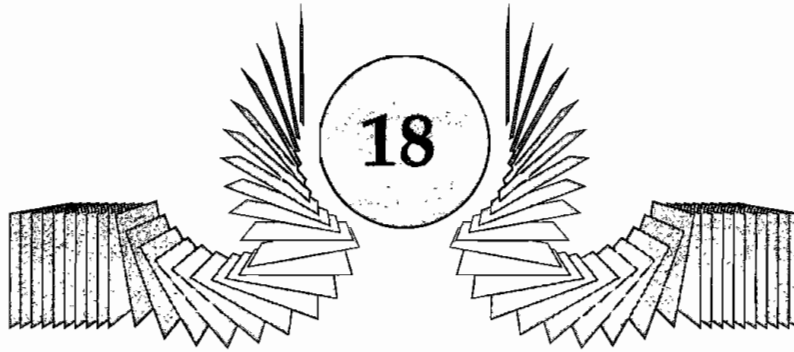
### Enabling the Agent

Turn this agent on like any other: choose **View+Design**, select **Agents** in the Navigator pane, select **Averaging agent** in the View pane, and click the **Enabled** button. (Double click the agent's name to change the time or frequency when the agent is run.)

## SUMMARY

- When workstations are configured to use an Domino server for Web page retrieval, any and all Web pages are stored in the Public Web Navigator database on the Domino server.
- The database contains various Views, Folders, and Navigators that are used to manage and organize the Web pages that have been retrieved by users and stored in the database.
- The built-in Purge and Refresh agents respectively are used to manage the size and timeliness of the documents in the database.
- The various configuring options for a Domino server are found in the server's Server document in the Public N&A Book, the Web Retriever Administration document in the Web Navigator database, and in the Location documents stored in each workstation's Personal N&A Book.





## **BUILDING WEB ACCESS INTO NOTES APPLICATIONS**

Although most users interact with Web pages via the Web Navigator databases, Web access can be built into any Notes application. Imagine a database that includes information about your competitors, and has built-in buttons to access the company's Home Page, retrieve up-to-the minute stock quotes for the company, or perform complex, Internet-wide searches for words related to that company.

The Web Navigator databases themselves also offer opportunity for customization; for example, you can modify the Public Web Navigator database to link to your own company's Home Page, or perhaps add customized Views to slice and dice the Web pages based on fields other than the defaults.

In this chapter you'll learn about including **@functions** and features into Notes databases to leverage Notes' ability to retrieve Web pages, as well as methods of customizing the Public and personal Web Navigator databases.

## INCLUDING WEB ACCESS IN OTHER DATABASES

As discussed in Chapter 16, one of the central tenets of Lotus' Web philosophy is that a user has access to Web pages anytime, anywhere. This promise extends to Notes database applications, in which you can include buttons, SmartIcons, agents, Hotspots, etc. that take advantage of Notes/Web integration.

### @URLOpen

No doubt the most important **@function** to consider when integrating a Notes database with the Web is **@URLOpen**. This is the function that designers use to incorporate Web access into a Notes application.

All by its Lonesome

In its simplest form, this function is used all by itself—in other words, without any arguments. When a Form's button, or a SmartIcon, or an Action bar's button includes this function all by itself, it produces the **Open URL** dialog box. For example, Figure 18.1 shows a Form from an example database; notice the **Open URL** button in the Action bar. When the user clicks this button, the **Open URL** dialog box appears, also visible in the figure.

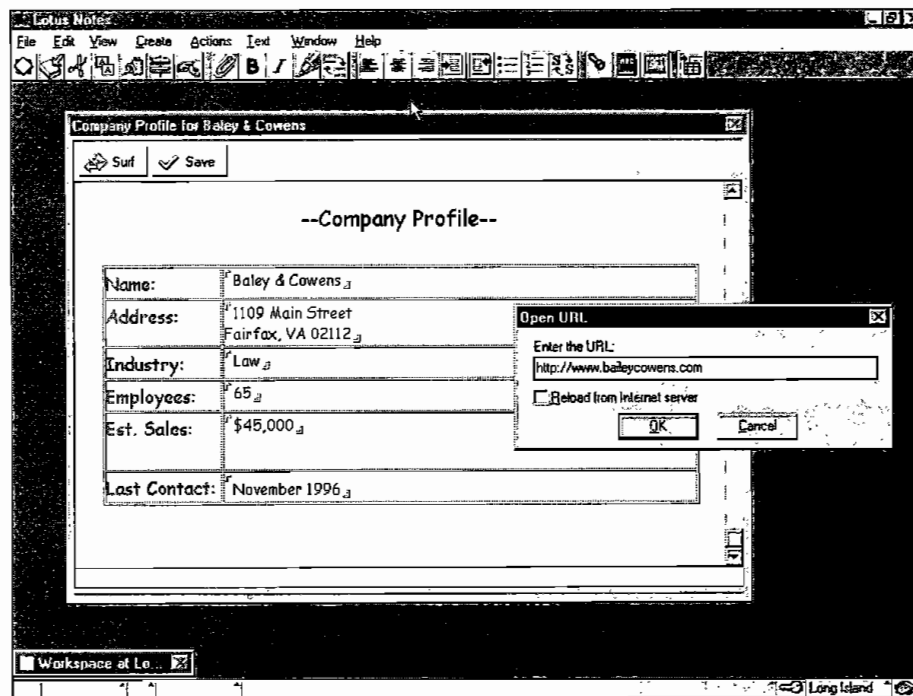


Figure 18.1 Using **@URLOpen** in a formula produces the Open URL dialog box.

Figure 18.2 shows how this Action bar was programmed—as promised, the button’s formula is nothing more than the **@function**, without any arguments.

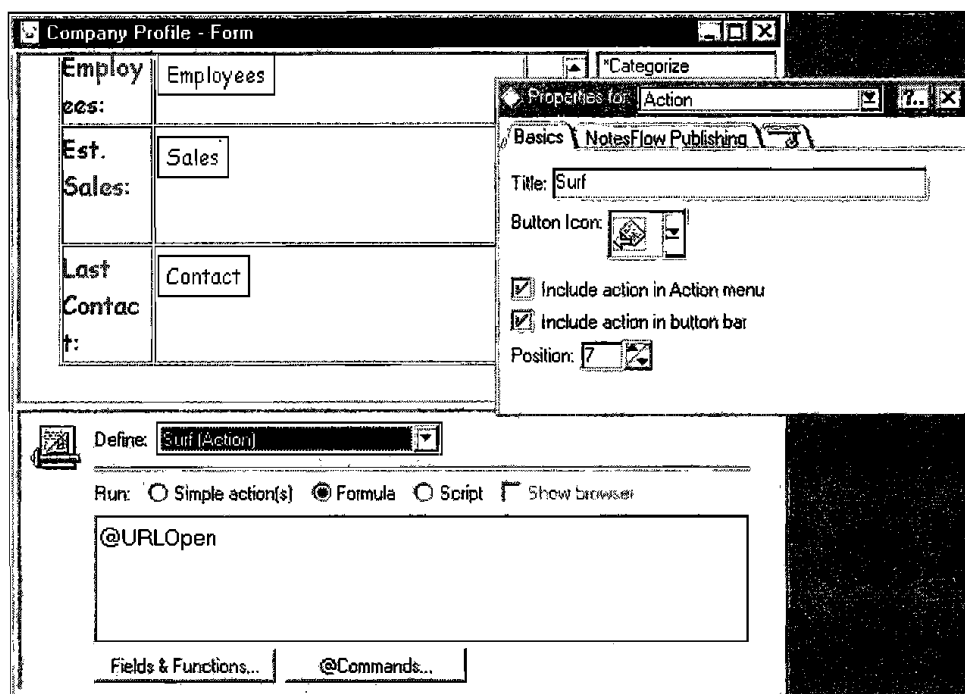


**Figure 18.2** Entering **@URLOpen** in an Action bar’s formula box.

### With an Argument

A slightly more complicated version of this function uses an argument for a specific Web page to be retrieved. For example, an Action Hotspot on a Form could be “hard-coded” to always open Web page on an organization’s internal, Intranet-based Web server for HR Policies and Procedures. When this function is used to open a particular page, the **@function** includes the name of the Web page to be retrieved in quotes, as in

```
@URLOpen("http://www.mispress.com")
```

or

```
@URLOpen("http://OurIntranetServer.com/HR_Policies.html")
```

Some Web servers require a user name and password before allowing access. The **@URLOpen** function supports accessing such pages via an

extended set of arguments designed specifically for this purpose. For example, imagine a SmartIcon's formula that included the function

```
@URLOpen("http://domino.lotus.com/domsite/30daydominodisc.nsf/";0;"Stephen
Londergan";"stevo")
```

Clicking this SmartIcon opens a discussion database on the Lotus Web site that requires a password for access. In this example, the second-to-last argument ("Stephen Londergan") is the user name, and the last argument ("stevo") is the password. For @URLOpen to succeed at opening a page that requires such a user name and password, the user name and password must have been already registered with the Web site in question.

(In the previous example, the two zeroes in the function are required arguments that are explained later.)

#### Substituting Field Values for an Argument

Figure 18.3 shows a document from a sample database that's used to track information about certain companies. In the table, there is a button that has been programmed to search the Web (using the AltaVista search engine) for Web pages that include the name of the company.

The screenshot shows a Lotus Notes application window titled "Company Profiles - All". The window has a menu bar (File, Edit, View, Create, Actions, Window, Help) and a toolbar. On the left is a "Folders and Views" pane showing "All", "Agents", and "Design". The main area contains a table with the following data:

| Company Name      | Industry   | Sales    |
|-------------------|------------|----------|
| Baird & Bartlett  | Paper      | \$21,000 |
| IBM               | Technology | \$700    |
| Jiffy Rentals     | Cars       |          |
| Lotus Development | Software   | \$32,780 |
| Microsoft         | Technology |          |
| MIS Press         | Publishing |          |

Below the table is a section titled "--Company Profile--". It contains a form with the following fields:

|            |                                             |
|------------|---------------------------------------------|
| Name:      | Lotus Development                           |
| Address:   | 55 Cambridge Parkway<br>Cambridge, MA 02142 |
| Industry:  | Software                                    |
| Employees: | 5000                                        |

To the right of these fields is a button labeled "Search the Web". The status bar at the bottom indicates "Workspace at Lo..." and "Long Island".

**Figure 18.3** Forms can include buttons using @URLOpen in conjunction with the database's fields.

Since the database includes profiles for many companies, “hard-coding” the formula for that search button with the name of a company simply won’t do. Instead, the button is programmed to use the contents of the `CompanyName` field as the basis of the search.

The actual function used in this example button is

```
@URLOpen("http://www.altavista.digital.com/cgi-bin/
query?pg=q&what=web&fmt=.&q=%22"+CompanyName+"%22")
```

The first step in programming such a button is to determine the *exact* URL that the engine expects when searching the Web for a page. The easiest way to do this is to perform a test search with a Web browser such as Netscape’s Navigator, and to then copy the resulting URL from the Search bar to the clipboard. Figure 18.4 shows the result of an AltaVista search; select the text in the Search bar and press **Ctrl+C** to copy it to the clipboard.

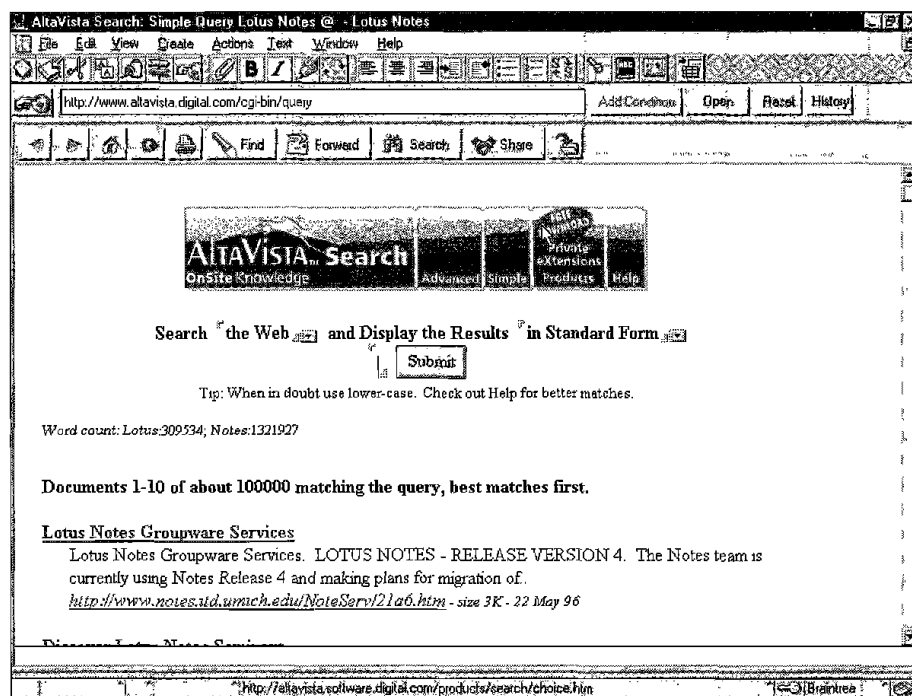


**Figure 18.4** Perform a test search to determine the exact URL that a particular search engine uses.

Once the correct URL is in the clipboard, paste it into the formula box for the button, Hotspot, etc. that you are creating. In the example shown in Figure 18.4, the search was conducted for the string `lotus notes`. Note that in this example, the AltaVista search URL ends up being

```
http://www.altavista.digital.com/cgi-
bin/query?pg=q&what=web&fmt=.&q=lotus+notes
```

The most important thing to note about the URL is that the search string is the last part of the URL; that's the part that will be replaced by a field name in the formula for the button in the database. Figure 18.5 shows the formula box containing the **@function** that is used in our example database to search the Web for the contents of the `CompanyName` field.

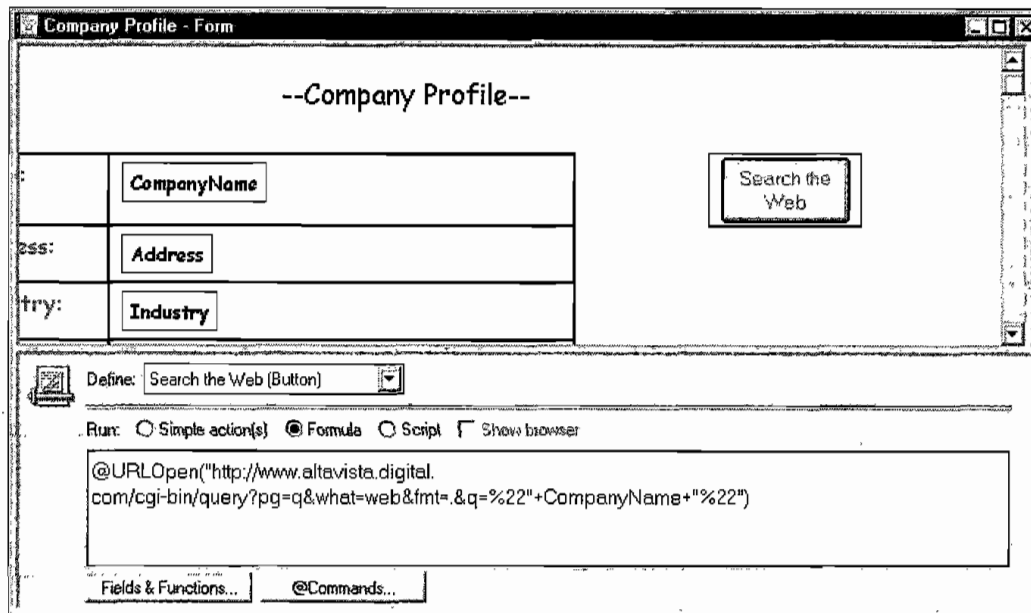


**Figure 18.5** Building a button that performs a dynamic Internet search based on the contents of a field.

The function programmed for the button is

```
@URLOpen("http://www.altavista.digital.com/cgi-
bin/query?pg=q&what=web&fmt=.&q="+CompanyName)
```

This function simply uses string concatenation to build a URL that includes the AltaVista's Search URL with the contents of the `CompanyName` field in the document.

### Using Other Search Engines

Although the previous example based a query on the AltaVista search engine, this is not meant to imply that this is the only search engine you

can use with Notes; any of the various engines (such as Yahoo, Lycos, etc.) will work. It's just a matter of determining the appropriate URL for the search engine you want to use, and then replacing search string with the name of a field from a Notes document. For example, the Notes formula

```
@URLOpen("http://www.lycos.com/cgi-  
bin/pursuit?cat=lycos&query="+CompanyName+"&x=28&y=1")
```

will do the same job as the previous example, but uses the Lycos Web search engine instead of AltaVista's.

It's a curious twist of fate that most of the Internet's search engines are not well prepared to handle spaces in a search string. For example, searching AltaVista for *lotus notes* actually uses the URL

```
http://www.altavista.digital.com/cgi-  
bin/query?pg=q&what=web&fmt=.&q=lotus+notes
```

Most search engines interpret a space as a Boolean AND; searching the Web for pages that contain the word "lotus" *and* the word "notes" will produce very different results from searching the Web for the *phrase* "lotus notes." When AltaVista (and most other search engines) search for a string that includes more than one word, the URL has to replace spaces with a plus sign (+).

At the risk of splitting hairs, this presents a problem for our example button, particularly in the case of a company whose name includes a space. The formula shown in the example would submit such a company name with the space intact, and the resulting query would employ the aforementioned Boolean operation. For example, if the company's name was "Lotus Development," the formula in the example would actually search for Web pages that had the word "lotus" anywhere in the page, and that also had the word "corporation" anywhere in the page—rather than searching for the phrase "lotus+development."

Fortunately, there is a Notes **@function** that solves this problem quite handily. The Notes function **@ReplaceSubstring** takes three arguments: the string to be searched, the string you're searching for, and the string with which the string you're searching for should be replaced.

For example, the formula

```
@ReplaceSubstring("lotus development";" ";"+"")
```

would return `lotus+development`, which is exactly the type of string that most Web search engines expect in order to search for a phrase.

The bottom line is that our example formula should really be

```
@URLOpen("http://www.lycos.com/cgi-bin/pursuit?cat=lycos&query="+@ReplaceSubstring(CompanyName;" ";"+"")+"&x=28&y=1")
```

This formula handles a company name that has spaces in it, and results in a Web search that more accurately searches for the entire company's name *as a phrase*.

#### Using @Prompt with @URLOpen

The last example uses the **@URLOpen** function in combination with **@Prompt** to create a SmartIcon that retrieves an up-to-the-minute stock price quote for a given stock value.

First, it was determined that the URL used to obtain a stock quote was

```
http://www.pcquote.com/cgi-bin/getquote.exe?TICKER=
```

followed by the four-letter stock symbol being sought. Since the button was designed to allow a user to enter a different stock value each time the formula was run, the button was programmed with the formula

```
Ticker:=@Prompt([OKCANCELEDIT]; "Stock Quote"; "Enter the company stock symbol in the box below.");
@URLOpen("http://www.pcquote.com/cgi-bin/getquote.exe?TICKER="+Ticker)
```

This formula uses the **@Prompt** function to display a dialog box in which the user enters the four-letter symbol for the stock. The value entered in the dialog box is assigned to the temporary variable **Ticker**. The formula then goes on to use string concatenation to combine the URL for the stock quote service with the value of the temporary variable **Ticker**. The result is a SmartIcon that, when clicked, prompts

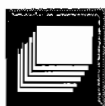


the user for a stock ticker symbol, and then uses the stock quote service at *www.pcquote.com* to obtain an up-to-the-minute stock price for the company, based on NYSE trading.

As was the case with the previous examples that illustrated formulas for Internet searches, **@Prompt** can be used in conjunction with any stock quote service's URL—or any other Web page that performs some sort of search or information retrieval operation based on an argument passed to it. Use **@Prompt** to provide a dynamic and interactive way for users to interact with the Web within a Notes database.

### **@URLOpen—Variations on a Theme**

Although the **@URLOpen** will accept several arguments that are intended to refine the way the function works, the good news is that, most often, the only argument needed for this function is the URL of a page to be opened. In the interest of total disclosure and accuracy, what follows is a complete explanation of this powerful function, along with its various and sundry arguments.



#### **NOTE**

Using the **@URLOpen** function in a formula mimics exactly a user using the choosing **File+Open URL**: specifically, the page that is retrieved as a result of the formula is stored in the Web Navigator database.

- **@URLOpen**: As we have seen, this is the simplest form of this function. When used by itself in a formula, this function produces the Open URL dialog box.
- **@URLOpen(URLString)**: The first argument that can be passed to the function is **URLString**, which is a text value representing the URL address of a page to be opened. This string can be a "hard-coded" URL address enclosed in quotes, or it can be the result of a string concatenation formula as illustrated in the previous examples.
- **@URLOpen(URLString;[reloadflag])**: Using the **[reloadflag]** argument determines whether or not the function will search the Web Navigator database to see if the page has already been stored there. Enter the string **Reload** to force the page to be retrieved from the Web server; use the string **ReloadIfModified**

to instead retrieve the page from the Web Navigator database. If the page being retrieved has not been previously stored in the database, the formula will retrieve it from the Web server, regardless of how you program this argument.

- **@URLOpen(URLString:[URLList]):** Most Web pages have Links on them, so the [URLList] argument is used to determine whether or not Notes should save the Links in a special field within the document. If you set this argument to the number one, Notes will store each of the links on the page being retrieved in a special field, named `URLLinks`<sub>n</sub>. (Since the individual fields can only be 64k bytes long, a page with a lot of links will have several of these fields: `URLLinks1`, `URLLinks2`, `URLLinks3`, and so on.) Using this argument to save the Links on a page that's been stored in the Web Navigator database is a way to provide access to those URLs; you could, for example, create an agent that retrieved a given page, *and* retrieved the pages to which the that page has Links. (This argument is the basis of the built-in Web Ahead agent in the Personal Web Navigator database.)

Use the number string `URLList` to save the URL addresses of a page's Links in the `URLLinks` fields; set the argument to zero (0) to not save the Link's URL addresses.

- **@URLOpen(URLString:[reloadflag]:[URLList]):** This version of the function combines both of the previous arguments. For example, the function

```
@URLOpen("http://www.mispress.com";Reload:0)
```

retrieves a Web page without checking whether the document has already been stored in the Web Navigator database, and does *not* save the page's Link URLs in the aforementioned `URLLinks` field. Note that the two arguments in this case are separated by a colon.

- **@URLOpen(URLString:[reloadflag]:[URLList];charset\$):** Web pages are retrieved based on certain character sets; for example, the character set ISO-2022-JP is used to map the text on a Web page to the Japanese character set, while the ISO-8859-1 set uses

the US English character set. In the extremely unlikely event that you need to override the character set that Notes is using to interpret a given page, enclose the name of the character in quotes for this argument.

- **@URLOpen(URLString;[reloadflag];[URLList];charset\$;Webusername\$):** If a page being retrieved by the formula needs a user name to be entered, use the fifth argument to include that text value in the formula.
- **@URLOpen(URLString;[reloadflag];[URLList];charset\$;Webusername\$;Webpassword\$):** Most pages that require a user name for access also need a password, so use the sixth argument for the function to include the password.
- **@URLOpen(URLString;[reloadflag];[URLList];charset\$;Webusername\$;Webpassword\$;Proxywebusername\$):** Some systems that employ a proxy server for Web access also require a user name for the proxy server itself; the **URLOpen** function can retrieve pages through such authenticated proxies by employing this seventh argument, which is a text string representing the user name for the Proxy server.
- **@URLOpen(URLString;[reloadflag];[URLList];charset\$;Webusername\$;Webpassword\$;Proxywebusername\$;Proxywebpassword\$):** Proxy servers that require a user name for access usually require a password as well, so the eighth text argument is provided to pass that value when a page is being retrieved.

One final note regarding this function: although the arguments for this function are optional, once you start, you can't stop. In other words, if your formula is to include a user name and password for the fifth and sixth arguments, you must include values for the second, third, and fourth arguments—even if you specify the defaults. For example, the function

```
@URLOpen("http://www.lotus.com";0;0;"";"stephen londergan";"stevo")
```

would retrieve the user name and password protected page at *www.lotus.com*. Notice that the default values have been included for the **[reloadflag]**, **[URLList]**, and that the **charset\$** argument was entered as a null string.

## Other @URL Functions

There are two other functions that can be used to incorporate Web access within a Notes database. Although they're included here in the interests of accuracy, it's worth mentioning that (a) they're seldom used by application developers, (b) they only work when used in formulas within the Web Navigator database, and (c) they're probably best left alone (unless you are bent on designing a completely customized Web Navigator database).

### @URLGetHeader

This function retrieves information about a given page, and is typically used only by advanced application designers. The types of information that can be retrieved by the function are based on HTML and industry standards. The function accepts up to six arguments, in the form of text strings. The syntax for the function is

```
@URLGetHeader (URLString;headerString;Webusername;Webpassword;Proxywebusername;Proxywebpassword)
```

Table 18.1 illustrates each of the six arguments.

**Table 18.1** @URLGetHeader arguments

| Argument         | Meaning                                                                   | Notes/example                                                                                                                                        |
|------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| URLString        | The URL for the page to be retrieved                                      | "http://www.tiac.net/users/stevo"                                                                                                                    |
| HeaderString     | The type of information to be retrieved about the page                    | "last modified", "server"; (Visit <a href="http://www.w3.org/">http://www.w3.org/</a> to learn the up-to-the-minute header string supported by HTML) |
| Webusername      | The optional user name required to retrieve the page                      | "stephen londergan"                                                                                                                                  |
| Webpassword      | The optional password required to retrieve the page                       | "stevo"                                                                                                                                              |
| Proxywebusername | The optional user name for the Proxy server required to retrieve the page | "stephen londergan"                                                                                                                                  |
| Proxywebpassword | The optional password required to retrieve the page                       | "stevo"                                                                                                                                              |

**@URLHistory**

The URLs that a user has visited during a “surfing session” are stored internally in a list, and are available for programmatic access via the **@URLHistory** function. This function is similar to an action button, in that using it actually displays a page from stored history. The function uses one argument, which is always enclosed in [square braces]. Table 18.2 shows the five possible ways the function can be used.

**Table 18.2 @URLHistory**

| Function              | What it Does                                                                       |
|-----------------------|------------------------------------------------------------------------------------|
| @URLHistory([next])   | Opens the next Web page in the history list                                        |
| @URLHistory([prev])   | Opens the previous Web page in the history list                                    |
| @URLHistory([show])   | Shows the pages in that the user has visited                                       |
| @URLHistory([save])   | Saves the current list of Web pages that a user has visited in a Web Tour document |
| @URLHistory([reload]) | Reloads the history list from the Web Tour document                                |

## CUSTOMIZING THE WEB NAVIGATOR DATABASES

The Public and Personal Web Navigator databases come pre-made with a collection of Views and Folders that are used to organize the documents that have been stored in them, but it is important to remember that the databases are just like any other Notes applications, and can be customized to include a myriad of special Views, Forms, agents, and so forth.

For example, you might add a new View to the Public Web Navigator database to sort and display the documents that have been retrieved today; or perhaps add a Folder that sorts and counts the documents based on the Web host server from which they were retrieved. Or maybe a Search agent that mails a newsletter summary whenever a document containing a certain phrase has been retrieved by a user? Likewise, it's conceivable to add a View to the Personal Web Navigator database that displays only the pages that contain a certain string within the page's URL.

Such customized database elements can use any of the “normal” Notes **@functions** (such as **@created**, **@Author**, **@ DocLength**, and so on).

## Special Fields

Web pages that have been stored in the Navigator databases contain all of the HTML content programmed by the page’s author, but Notes also adds a few fields that can help you sort and manage the pages. Refer to Table 18.3 when building a custom Form, View, or agent for these special fields’ names and purposes. Each of these fields is created automatically when a Web page is stored in the Navigator database, and each can be used in any button, action or agent in the database.

**Table 18.3** Special HTML fields

| Field name  | Purpose/example                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Host        | Used to store the name of the Web server from which the page was retrieved/ <i>www.mispress.com</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| URL         | Stores the exact URL for the page/ <i>http://www.mispress.com/marketing/hothap.html</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Title       | Stores the title of the page, directly from the HTML header tags within the page/Microsoft Investor: Quote Lookup                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| PrivatePage | Indicates (via the number “1” or “0”) whether or not the page is private                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| URLLinks#   | Stores a list of all the URLs on the page. (Since the field can only contain 64k, there may be more than one of these fields.) For the page at <i>www.ibm.com</i> , the field named <i>URLLinks1</i> would contain:<br><br>“ <i>http://www.ibm.com/News/</i> ”<br>“ <i>http://www.ibm.com/Products/</i> ”<br>“ <i>http://www.ibm.com/Services/</i> ”<br>“ <i>http://www.ibm.com/Links/http://www.internet.ibm.com/</i> ”<br>“ <i>http://www.ibm.com/Solutions/</i> ”<br>“ <i>http://www.ibm.com/IBM/</i> ”<br>“ <i>http://www.ibm.com/trademarks.html</i> ” |
| HTTPHeaders | Stores a list of all the HTTP header fields on the page. Each header field is stored as a separate field with its own value. For example, <i>HTTPLast_Modified</i> holds the date on which the page was last updated on the Web server.                                                                                                                                                                                                                                                                                                                     |

|                |                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeferredObject | If the database's property is set to <b>Display images after loading</b> , all of the objects (GIFs, JPEGs, Java applets, and so on) are stored in separate documents within the database. This field contains a "1" or a "0" to indicate if the document is one of these deferred objects that has been stored separately from the original Web page.                               |
| HTTPStatus     | Stores an HTTP status code, representing the status of the Web page retrieval. For example: status code 200 means a successful request and retrieval of the page. (All of the possible codes and values for this field—and information about other HTTPHeader fields—are available at <a href="http://ds.internic.net/rfc/rfc1945.txt">http://ds.internic.net/rfc/rfc1945.txt</a> .) |

## Custom Search Agents

Since the Public Web Navigator ends up storing *all* of the Web pages that have been visited by users via the Domino server, it becomes a powerful tool for searching in and of itself.

A popular and particularly useful custom agent for this database is one that searches the database for a particular string or phrase, and then sends a newsletter summary by e-mail to notify you (or, perhaps, a group of people) when a page meeting those criteria has been added to the database. This agent could also be used in a Personal Web Navigator database in conjunction with the Page Minder and Web Ahead agents.

### Creating a Custom Search Agent



#### NOTE

This agent—and any other agent that's intended to search the Web Navigator database—requires that the database have an up-to-date Full-text index.

This agent is designed to run once each day, at the default time for daily agents of 1:00 A.M. As shown in Figure 18.6, the agent is designed to find documents that include the words **new** and **insurance** and **regulation**. And as specified in the agent's formula box, the agent does two things when it finds documents meeting the search criteria: first,

the agent prepares a newsletter summary of the documents, and mails it to the person included in the formula box. Second, the agent moves the document to the Web Ahead folder, so that linked pages on the page including the Search criteria will be automatically loaded by the Web Ahead agent.

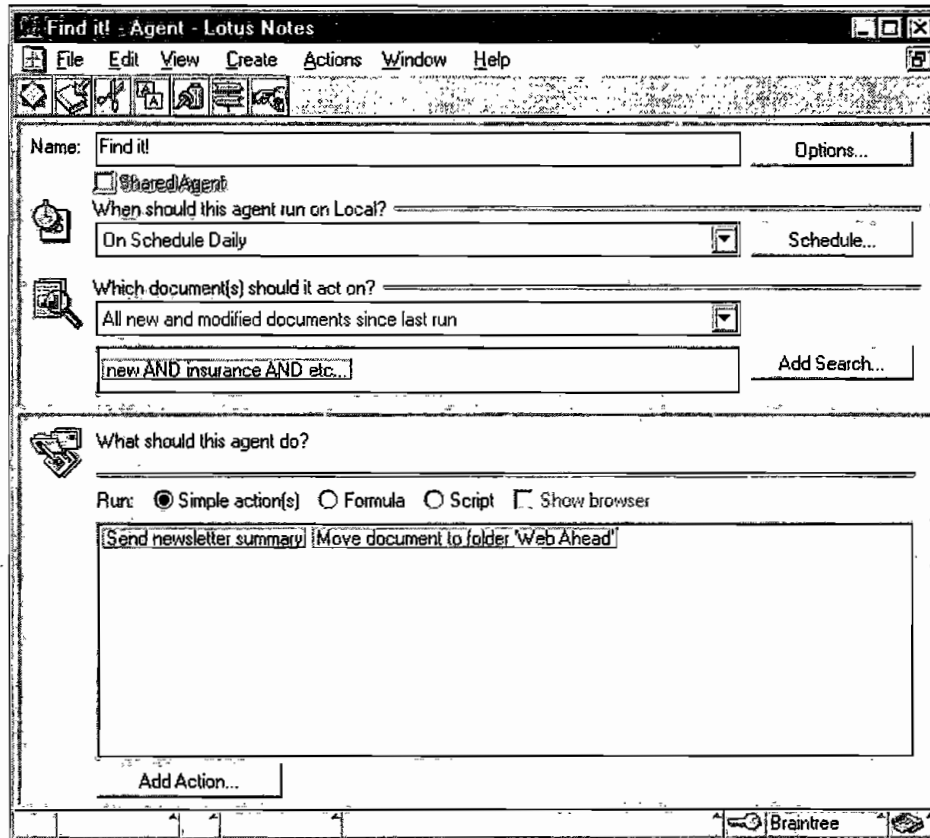


**Figure 18.6** This simple agent regularly searches the Web navigator database for the words “new insurance regulation” and then mails a newsletter summary of the documents it finds, as well as moving the documents to the Web Ahead folder.

As shown in Figure 18.7, there’s quite a bit of functionality lurking in the Add action button; other agents in the database could be created to automatically delete documents, or perhaps even move the documents to a different database altogether.



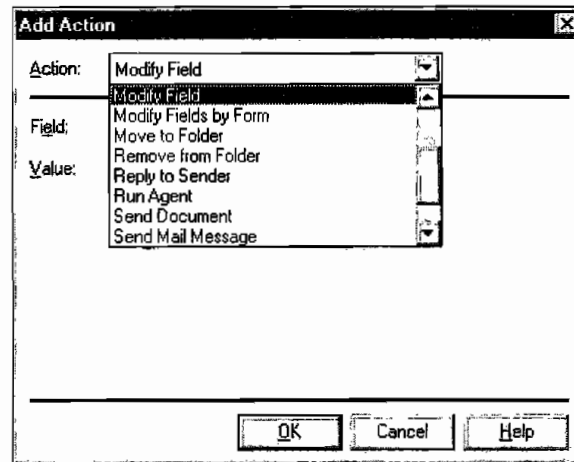


**Figure 18.7** An agent can perform many, multiple actions on a document.

### Some Other Ideas for Custom Web Agents

As you might suspect, there's virtually no end to the possibilities that can be programmed with a database agent, and running agents on the Web Navigator database provides some truly awesome opportunities for integrating Web pages with e-mail, workflow, and other Notes applications. Also note that the Web Navigator database agents fully support LotusScript.

A few suggestions for agents that showcase Notes integration with Web pages follow—consider an agent that:

- Retrieves a stock price for a given stock symbol every hour, and mails the document to a group of people;
- Opens and retrieves the daily "Dilbert" comic strip each day and mails the document to the database's owner;
- On a weekly basis, refreshes the documents in the database that have attachments, to ensure that the files are kept up to date;
- Regularly opens and mails a Web page that has a JPEG image of the current weather conditions in Stockholm;
- Searches and collects Usenet posts about a particular topic, competitor, or product.

## ONE LAST (COOL) TRICK

The last piece of functionality in Notes is the ability to create a Form that auto-launches a URL. By using a special text field name and a Form Property, you can create a document, that when opened, automatically retrieves the Web page's whose address is stored in the aforementioned special text field.

The first \_ of the secret to this auto-launching Form is a special field; although the contents of the field can be editable or computed, the field must be named **URL**. The second \_ of the secret is contained in the **Launch** tab of the Form's **Properties** box, where you choose **URL** from the Auto Launch list.

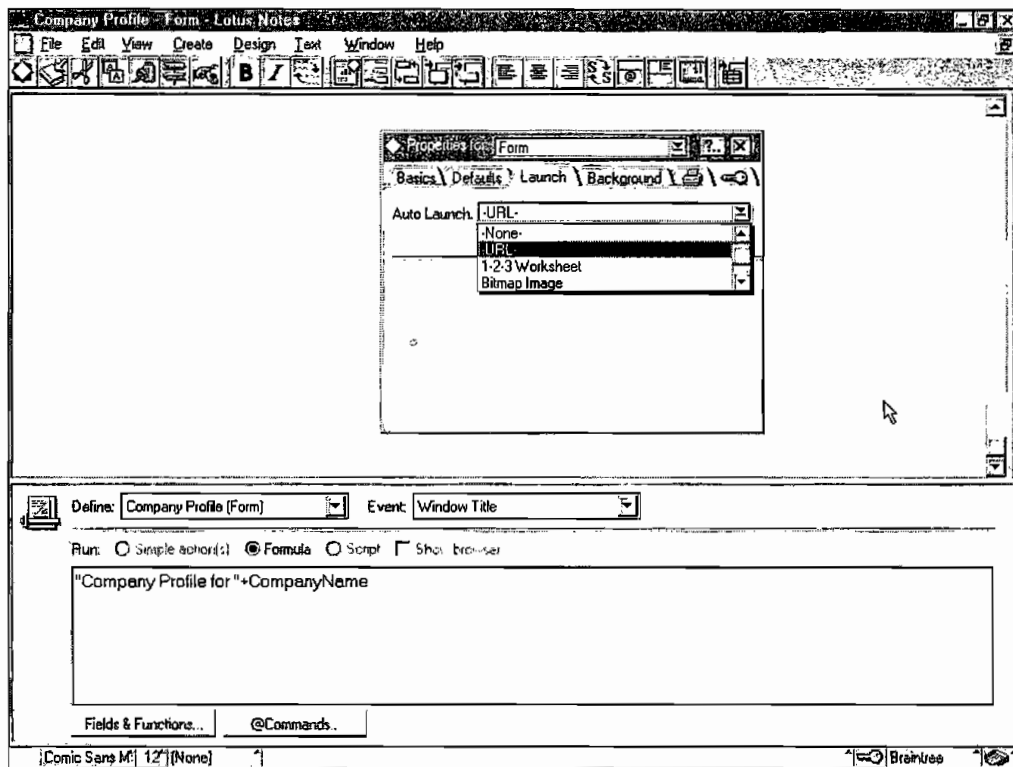


**Figure 18.8** Set a Form's **Launch** property to **URL** to have the Form automatically retrieve and display a Web page.

Whenever a Form with such a field and Launch property is opened, Notes automatically retrieves the Web page at the address stored in the URL field.

Imagine a database of profile documents for employees at a company—one listing information about the employee, one listing their last performance appraisal, and the last automatically opening the person's Home Page!

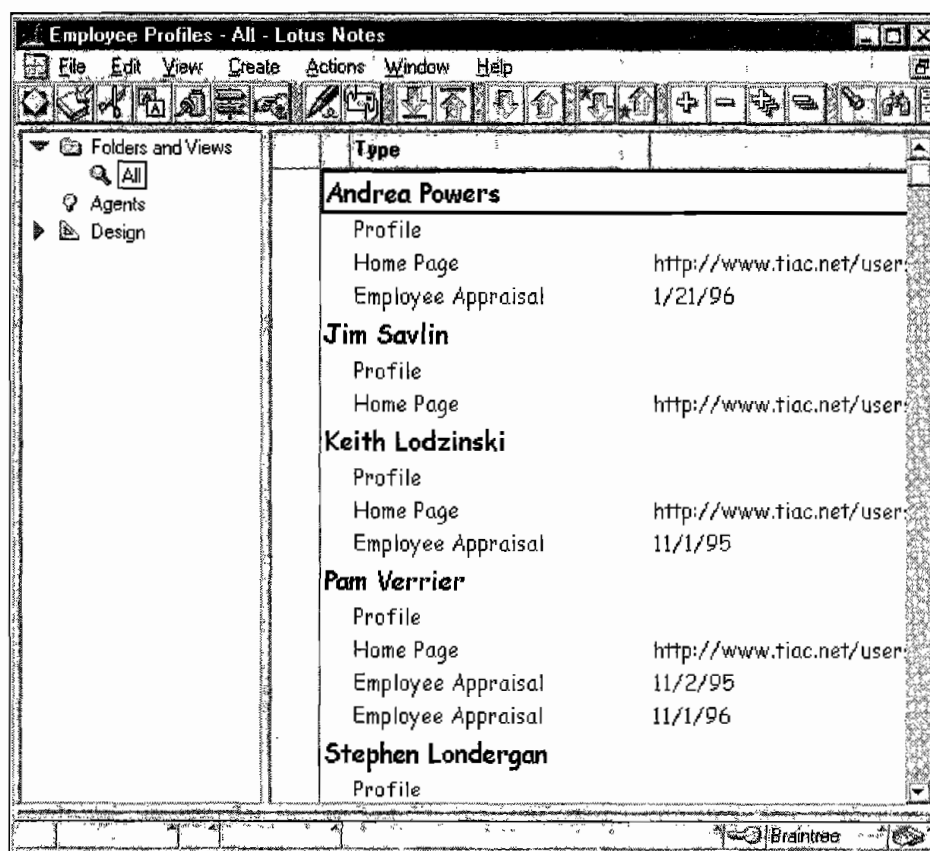
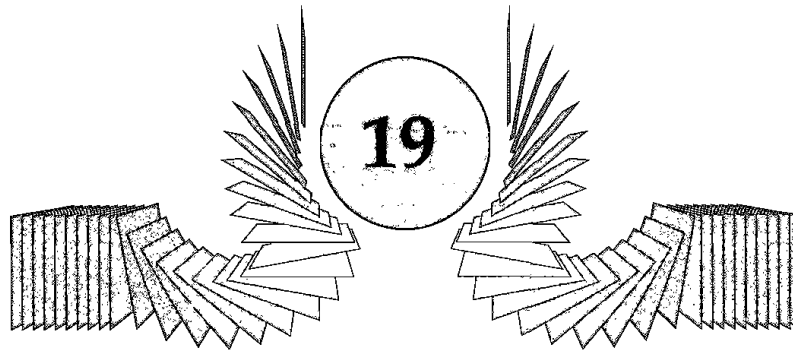


Figure 18.9 A database using a URL Autolaunch Form.

## SUMMARY

- Access to Web pages and FTP sites on the Internet and Intranet can be integrated into any Notes database application.
- Agents can lend powerful Web page retrieval, search, and “worming” features to Notes applications.
- Notes contains several @Functions that are used for Web page retrieval, the most important of which is @URLOpen.



## MISCELLANEOUS SURFING ISSUES

Not all of the topics related to browsing the Web with through a Notes workstation fit nicely into the other chapters in this part of the book, so Chapter 19 lists a few unrelated, but important, surfing issues.

In this chapter, you'll learn about how the Notes workstation browser handles Java applets, running Netscape Plug-ins, configuring multiple Location documents for a single Notes workstation, and configuring the workstation's browser to access the Internet Proxy server.

### JUICED ON JAVA

Java programs—usually called *applets*—are small programs that are contained within Web pages. The idea behind a Java applet is that the program is small enough to be downloaded on an as-needed basis, by each user who loads the Web page. The spinning globes, animated icons, and flashing banners that have become so prevalent on Web pages lately are most always the result of one of these small programs.

When a Web browser encounters a Web page that includes the HTML `<APPLET>` tag that is used to indicate a Java applet, the applet's code is temporarily copied to the workstation's memory and then executed.

In addition to being small, Java programs are written in a language that most Web browsers (including Notes) can interpret, compile, and run. This is a great boon to the Java programmer, who can be certain that his (or her) program will run on just about any system, without having to worry about the type of computer or operating system that the Web user will have. The good news is that the Web browser built-in to the Lotus Notes 4.5 workstation supports Java fully, so a Notes user will experience the full effect of the Web while surfing.

But there is a hidden threat in this Java business: most network administrators shudder at the thought of the casual Web surfer downloading and running programs via Web pages. After all, there are bad people in the world who might create a seemingly innocuous applet that in fact mimics a virus in its ability to wreak havoc on the user's workstation and network. (But an applet running in the Notes workstation can never access the user's local hard disk, password files, and so on.) What's even worse is that the user may not even know that the page they're about to load contains an applet all.

Fortunately, the Notes workstation has several options which can be fine-tuned to control exactly what an applet can or can't do—thereby limiting the potential impact of a “bad” applet. This feature is of particular interest to the Intranet administrator, who would presumably trust applets created within his or her organization, but who would be obviously more suspicious of those originating on foreign Web sites.

### **Running Applets on a Notes Workstation**

When the Lotus Notes 4.5 workstation software is installed, there are certain files that have to be copied for applet execution—and although the default is that these files are automatically installed along with the rest of the workstation software, reinstalling them (or at least confirming their installation) is often the first step when troubleshooting applet execution problems.

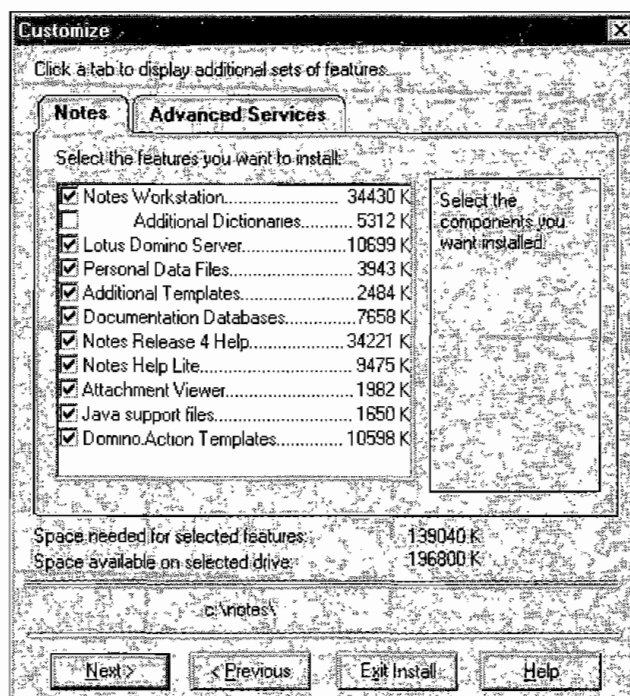


**Figure 19.1** To run Java applets, a workstation has also got to have Java execution enabled, via the **File+Tools+User Preferences** dialog box.

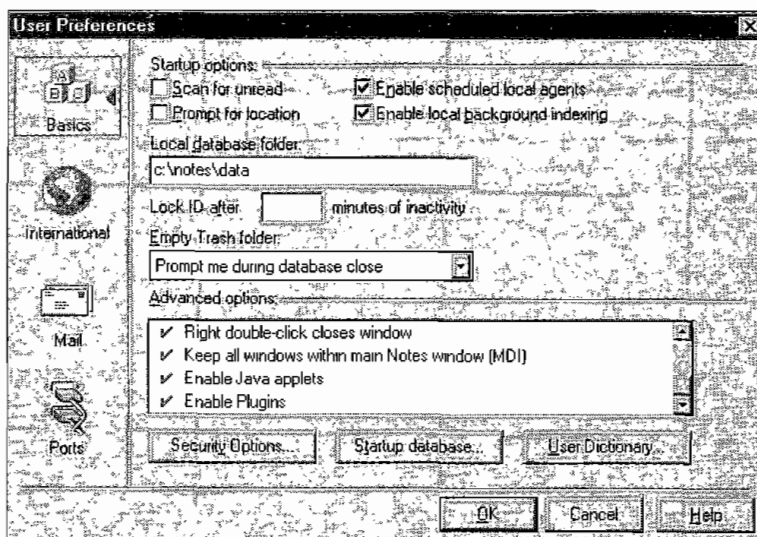


**Figure 19.2** Use the Java Console to see what an applet has been up to.

Finally, every Notes workstation has a *Java console* that can be used as an additional troubleshooting tool if applets aren't executing properly. This small window shows the current status of any applets that are running; the idea is to open the Web page on the workstation that contains the troublesome applet, and to then choose **File+Tools+Show Java Console** to see what's going on.

## Controlling Java Applet Execution

Since Java applets can pose a security risk by exposing the workstation and its network to the "outside" Internet, the most important tool available to the Administrator regarding a workstation's ability to execute Java applets is the **Java Applet Security** section of a user's Location document. Table 19.1 illustrates each of the fields, along with their possible values.

**Table 19.1** Java Applet security settings

| Field Name                       | Possible Values                                                                 | Comment                                                                                                                                                                                                                                            |
|----------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Trusted Hosts                    | Any combination of Host names, IP addresses, and wildcards, separated by commas | The systems listed here will be able to load Java applets on the user's computer. In an Intranet, the Trusted Hosts would probably be those belonging to "your" company.                                                                           |
| Network access for Trusted Hosts | Disable Java                                                                    | This field determines what the Trusted Hosts can do on the Trusted Hosts user's computer. <b>Disable Java</b> in particular prevents any of the locations listed in the <b>Trusted Hosts</b> field from running Java applets on the user's system. |
|                                  | No access allowed                                                               | A Trusted Host can run applets on the user's system, but the applet can not make <i>any</i> HTTP connections.                                                                                                                                      |
|                                  | Allow access only to originating host                                           | A Trusted Host can run applets on the user's system, but the applet can make HTTP connections only to the Host from which the applet was retrieved in the first place.                                                                             |
|                                  | Allow access to any trusted host                                                | A Trusted Host can run applets on the user's system, and can make HTTP connections to any other of the Trusted Hosts.                                                                                                                              |
|                                  | Allow access to any host                                                        | A Trusted Host can run applets on the user's system, and can make HTTP connections to any other Host (trusted or not).                                                                                                                             |

|                                    |                                      |                                                                                                                                                                                                                                                                                                                      |
|------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network access for Untrusted Hosts | Disable Java                         | This field determines what the Hosts that are <i>not</i> listed in the <b>Trusted Hosts</b> field can do. Specifically, the <b>Disable Java</b> option means that the Untrusted Hosts can not run any applets on the user's system.                                                                                  |
|                                    | No access allowed                    | An Untrusted Host can run an applet, but can't make HTTP connections on any other host.                                                                                                                                                                                                                              |
|                                    | Allow access to any originating Host | An Untrusted Host can run an applet, and can make HTTP connections only to the Host from which the applet was retrieved in the first place.                                                                                                                                                                          |
| Trust HTTP Proxy                   | Yes/No                               | This field determines whether or not the user's workstation is allowed to resolve Host names through their local Proxy server. (Some applets require Host name resolution for proper execution; this field works in conjunction with the <b>Web Proxy</b> field in the user's Location document). The default is No. |

## NETSCAPE PLUG-INS

Plug-ins are similar to applets in that they are typically small programs that are used to deliver customized content along with a Web page to the user. Two examples of popular Plug-ins include RealAudio, (<http://www.realaudio.com>) for playing streamed audio content, and Macromedia's (<http://www.macromedia.com>) Shockwave for delivering and displaying streamed audio, live video, and other such multimedia "experiences."

Plug-ins differ from applets in that the code that's needed to hear the audio, play the movie, and so forth, is installed on the user's workstation; unlike a Web page that has a Java applet that includes both the code and content, a Plug-in's code resides on the workstation, while the content resides (or is *embedded*) in the Web page itself. The downside to Plug-ins is that the program necessary to "play" the Web page has to be installed before the content is loaded; this often means that a user who encounters a page with Plug-in content has to stop, download the appropriate Plug-in software, and then return to the Web page that had the Plug-in's content in the first place.



The Notes browser fully supports the use of Plug-ins; whenever the browser encounters a Web page that includes the `<EMBED>` HTML tag that denotes Plug-in content, Notes assumes that the Plug-in associated with the embedded content is installed on the workstation.

Plug-ins were essentially invented by Netscape, and consequently the code for most of them is installed in the workstation's Netscape directory. When the Notes browser encounters a Web page with the `<EMBED>` tag, it searches the workstation's Netscape directory for the code required to process the embedded content. If the Plug-in is not found, Notes will prompt the user for the location on the workstation for the Plug-in code; if the Plug-in program has not been installed on the workstation, the user will (obviously) not be able to see/hear/play the Plug-in content—and presumably would then pause their “surfing session” to install the Plug-in code.

## MANAGING MULTIPLE LOCATIONS

A user's Location documents, which are stored in the workstation's Personal N&A Book, are used to configure the manner in which Web pages are retrieved on that workstation, at that location. Some users—particularly those with laptops—require that the Internet Browser and retrieve/open pages fields be configured differently to support the different ways in which they will be connecting to the Internet. For example, a workstation user with a laptop typically uses the InterNotes server when connected to the corporate LAN to retrieve Web pages; as discussed in Chapter 14, using a centralized InterNotes server offers several advantages in terms of network traffic and disk space savings.

At the same time, the user may have occasion to use the Notes workstation to retrieve Web pages—particularly to take advantage of the Page Minder and Web Ahead agents in the Personal Web Navigator database.

That same user, however, might connect to the Web differently when the workstation is not connected to the corporate LAN; specifically, the owner of the machine might connect to the Internet from home, sometimes by dialing in to the Notes server—using the Notes workstation to retrieve Web pages—and other times via a personal, dial-up Internet account with a local Internet Service Provider, which, for the sake of the argument, necessitates the use of either Netscape's Navigator or Microsoft's Internet Explorer.

Finally, the same user might use this laptop system while completely disconnected—in which case the machine would be configured to not retrieve Web pages at all. (Although its owner could still simulate surfing the Web by reading the pages in their Personal Web Navigator database, especially if the Web Ahead agent had previously been hard at work worming pages.)

Each of these five scenarios can be handled rather nicely by the Location documents in a user's Personal N&A Book. As we've seen, the fields in the Internet Browser section of a Location document determine how Web pages are retrieved on the workstation when a particular Location document is employed. In the case of the aforementioned traveling laptop, the Personal N&A Book would contain at least four distinct Location documents, as indicated in Table 19.2.

**Table 19.2** Internet Browser settings

| <b>Suggested Location Document's Name</b> | <b>Internet Browser</b> | <b>Retrieve/open Pages</b> | <b>InterNotes server</b>                      | <b>Comment</b>                                                                                                     |
|-------------------------------------------|-------------------------|----------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Office                                    | Notes                   | from InterNotes server     | enter the name in the InterNotes Server field | Uses the Public Web Navigator database—Web Ahead and Page Minder agents not available                              |
| Office with Agents                        | Notes                   | from Notes workstation     | N/A                                           | Web Ahead and Page Minder agents available via the Personal Web Navigator database                                 |
| Home via Notes                            | Notes                   | from InterNotes server     | enter the name in the InterNotes Server field | user would first dial in to the Notes server                                                                       |
| Home with Agents                          | Notes                   | from Notes workstation     | N/A                                           | Web Ahead and Page Minder agents available via the Personal Web Navigator database; assumes a connection to an ISP |
| Home via ISP                              | Netscape Navigator      | N/A                        | N/A                                           | Could also be configured to use Microsoft's Internet Explorer, or any other Web browser                            |
| Disconnected                              | N/A                     | No retrievals              | N/A                                           | Previously-retrieved pages are still available in the Personal Web Navigator database                              |

**NOTE**

Create Location documents in a workstation's Personal N&A Book by choosing **File+Mobile+Locations** and then using the Action bar's **Add** and **Edit** buttons.

**TIP**

The Administrator who has the luxury of using or not using a Proxy server to connect to Web sites can also employ separate Location documents with separate settings for the **Web Proxy** field to switch back and forth between using not using the Proxy server.

## CONFIGURING A WORKSTATION OR SERVER TO USE A PROXY SERVER

Both a Notes workstation and the Domino server can be configured to use a Proxy server—or servers—when connecting to Internet/Intranet hosts. In the case of a workstation that is configured to retrieve pages itself (without using the Domino server), Proxy configuration is done via the workstation's Location document. Proxy server configuration for a Domino server that retrieves Web pages for workstation users is accomplished via the fields in the **Proxy Configuration** section of the Server document in the Public N&A Book. In the case of multiple Location documents to support mobile users, as discussed in the previous section, each of the Location documents would have its own Web proxy settings in the Personal N&A Book.

To configure a Location document to access Web sites via a Proxy server, enter the Proxy server's IP address (or, if the workstation has access to a domain name server, its Host name) in the **Web Proxy** field in the **Basics** section of the Location document or Server document.

In some cases, different Proxy servers are configured, depending on the type of Internet service that will be accessed. In this case, use the "propeller beanie" button at the right of the **Web Proxy** field to enter the Host names/IP addresses of the other Proxy servers.

Figure 19.3 A sample Location document from a user's Personal N&A Book.

Figure 19.4 Configuring a workstation to retrieve Web pages via a Proxy server.

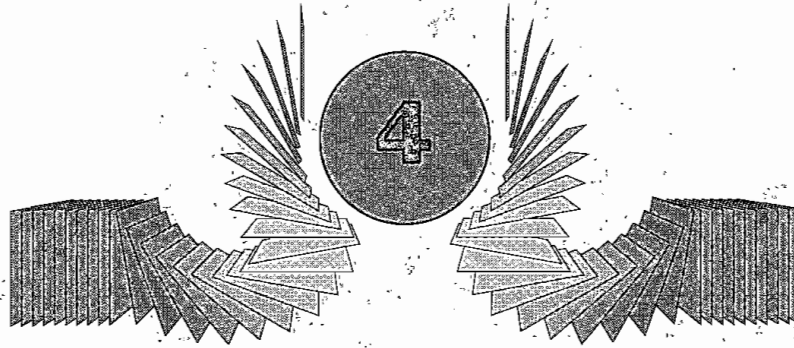
The Proxy Server Configuration dialog box and Server document make provisions for up to seven different types of Proxy servers:

- HTTP, used to configure the Host name or IP address of the Proxy server that the workstation will use to retrieve Web pages;
- FTP, used to configure the Host name/IP address of the Proxy used to retrieve files being downloaded via FTP;
- Gopher, used to configure the Host name/IP address of the Proxy for Gopher retrievals;
- SSL, used to configure the Host name/IP address of the Proxy used for Secured Socket Layer retrieval of secure Web pages;
- SOCKS, used to configure the Host name/IP address of the Proxy or SOCKS-based connections;
- Notes RPC, configures the Host name or IP address of the Proxy server used to connect to Notes servers via the Internet;
- No Proxy..., used to configure the Host names and/or IP addresses that can be connected to without using a Proxy. (In the case of workstations or servers that connect to both Internet and Intranet-based Web servers, the IP addresses/Host names of the Intranet servers are typically entered in this field, as they are presumably "safe enough" to not require the use of a Proxy server.) This field can also include wildcards, as in \*.lotus.com or \*.mispres.com.

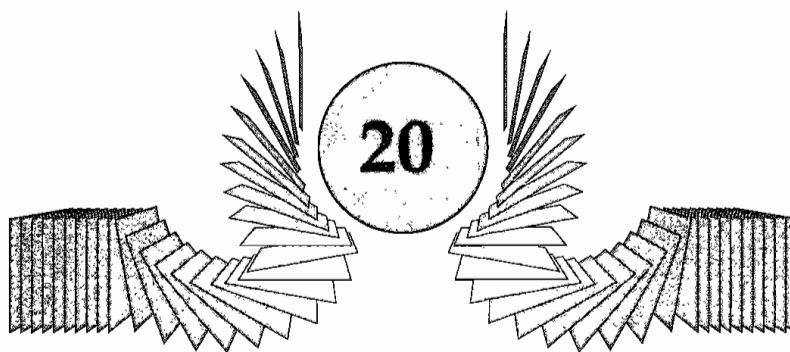
## SUMMARY

- The Notes browser fully supports Web pages that contain Java applets—provided that the workstation has been configured to run them via the File+Tools+User Preferences dialog box.
- Plug-ins are also supported; the Notes workstation uses the Netscape directory to find Plug-in programs.
- Configure Notes to use Proxy servers via a workstation's Location document in the Personal N&A Book or the Domino server's Server document in the Public N&A Book.

**SECTION**



**NEWS**



## **THE BIG PICTURE: NEWS & NOTES**

Given a Notes server that has a TCP/IP connection to a Usenet server, Notes users can participate in what has been called the “party” of Usenet. Part Four of this book discusses the Lotus Notes InterNotes News Gateway, which is a program that runs on a Domino server and automatically and regularly connects to Usenet servers to upload and download Usenet articles. In this chapter, you’ll get a brief introduction to Usenet, and an overview of how the Lotus Domino InterNotes News Gateway works.

### **WHAT IS USENET?**

Usenet is a lot like a huge system of computer bulletin boards, with an estimated 40 *million* people talking (and typing) about thousands of topics. It can be difficult to fathom the depth and breadth of information that is available on Usenet. With more than 40 *thousand* worldwide discussion topics that range from skiing to astrophysics to a David Letterman fan club, there is, for all practical purposes, no end to the discussion topics flowing daily through Usenet. At its best, Usenet gives you the opportunity to ask a question about any topic and have experts

from around the world respond with their timely and sage advice. At its worst, Usenet is, to paraphrase syndicated columnist Dave Barry, “a lot like CB radio with typing.” It has been estimated that Usenet users worldwide produce at least one gigabyte’s worth of new Usenet messages *each day*. Even if the origin, use, deployment, and configuration of Usenet is a topic more deserved of an entire book, there are a few key concepts and vocabulary words that you’ll need before planning and implementing the InterNotes News Gateway.

### So Much Information...

The foundation of Usenet is the *article*, which is a message that has been created and submitted by a person somewhere in the world. To provide some level of organization and control to the millions of Usenet articles produced every month, articles are organized into *newsgroups*. A newsgroup is, quite simply, a collection of articles about the same topic. If you were looking to read up on *The Hitchhiker’s Guide to the Galaxy*, you’d start with the newsgroup named *alt.fan.douglas-adams*, and not the newsgroup that’s named *alt.cars.ford-probe*.

There are more than 40,000 of these newsgroups, with new ones being created every day. Given so many, they are also arranged together, into *hierarchies*. A hierarchy is a collection of newsgroups that are somehow related. Since just about anybody in the world with the proper access to a Usenet server can create a new newsgroup, and since the hierarchy of a newsgroup is determined by the name of the newsgroup, it’s hard to estimate, on any given day, exactly how many hierarchies there are in existence. The most common hierarchies, sometimes referred to as “the big eight,” are:

- comp.\* for the discussion of computer-related topics
- news.\* for the discussion of topics related to Usenet itself
- rec.\* Recreationally oriented
- sci.\* for scientific discourse
- soc.\* for socially oriented discussions
- talk.\* for the “grand debates,” such as gun control, abortion, nuclear power, etc.



- misc.\* Miscellaneous
- alt.\* known as “The Altnet,” this hierarchy is responsible for 50% of the total daily Usenet article traffic. There are thousands of newsgroups in this hierarchy. This hierarchy contains the newsgroups that can be the most problematic for administrators in the workplace, such as *alt.sex*, etc.

Figure 20.1 shows a graphic representation of how this all fits together.

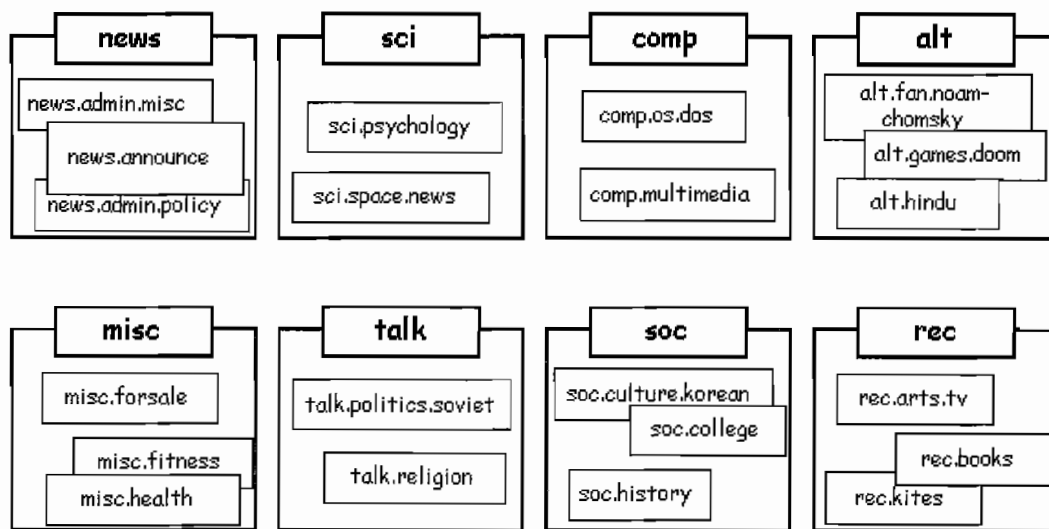


**Figure 20.1** Usenet articles are organized into newsgroups, and newsgroups are organized into hierarchies.

## Distribution

Users and administrators choose which newsgroups (or entire hierarchies of newsgroups) they want. A person is certainly incapable of following every last newsgroup in the world, and a company undoubtedly does not have the computing resources, disk space, and business need that would be required to subscribe to them all, either.

For example, let's say that Diane pays an Internet Service Provider (ISP) for access to the Internet. In addition to access to the World Wide Web and e-mail, Diane's ISP has a computer that regularly receives Usenet articles (called, for the record, a Usenet *server*, and also

occasionally referred to as a Usenet *host*). Diane can read articles that have been posted by Usenet users around the world, and she can submit (or post, to use the proper Usenet parlance) her own articles, too. Most ISPs provide access to just about all of the message hierarchies that are in existence, so Diane can read articles in any of the extant newsgroups. Of course, Diane probably interacts with a very small percentage of the available newsgroups—in fact, she may read the articles that are posted to just three or four.

Jack, on the other hand, has access to Usenet through his employer, who has set up a Usenet host of his own. Jack's employer has chosen not to subscribe to all newsgroups, so Jack doesn't have access to all of them like Diane has. (As we have seen, most companies would not bother subscribing to all of the Usenet newsgroups.) The Usenet administrator at Jack's company has chosen to ignore the entire *alt.\** hierarchy. See Figure 20.2.

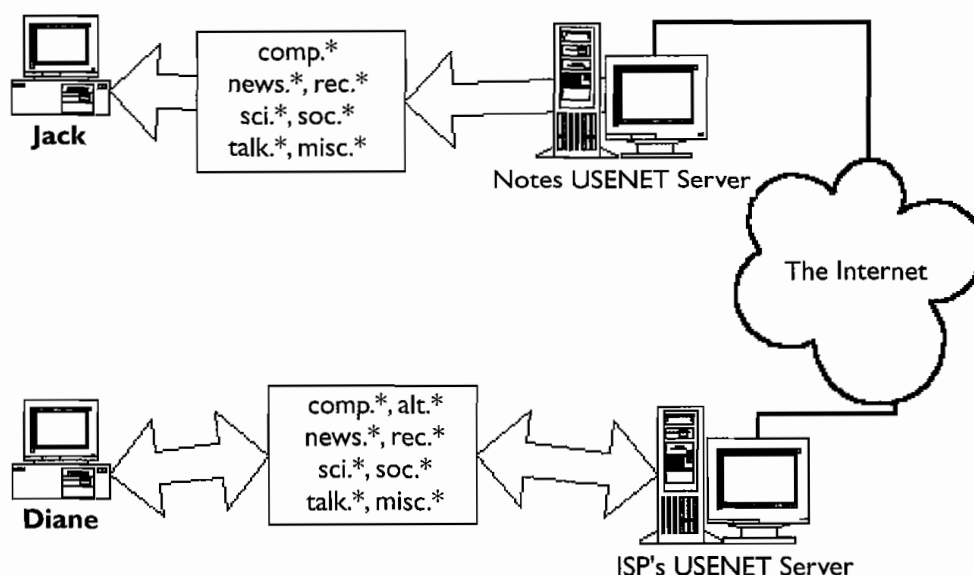


**Figure 20.2** Some Usenet users and servers do not have access to all of the estimated 20,000 news groups.

The point of this little ditty about Jack and Diane is that when the Usenet administrator is configuring the regular exchange of Usenet

articles between his server and some other, an important part of this job is determining which articles to exchange. When choosing which newsgroups or hierarchies of newsgroups his Usenet server will exchange, the Usenet administrator considers such factors as:

- Disk space—The more subscribed newsgroups, the more disk space used.
- Employee productivity—Should employees allowed to be reading newsgroups such as *rec.arts.disney.parks* during the workday?
- Relevance—Unless the company does business down under, why bother subscribing to *aus.jobs.wanted*?
- Propriety—There are certainly moral and perhaps even legal implications to subscribing to newsgroups of the *alt.sex* or *alt.binaries.pictures.variety*. It's up to the administrator to decide if an employee has or needs access to such newsgroups at the office.

The decision on which server your own Usenet server is going to exchange articles is important. If you and your users need access to *biz.marketplace.international*, you obviously need to exchange articles with a Usenet server that subscribes to that newsgroup. The Usenet servers maintained by universities and commercial Internet Service Providers tend to provide access to all of the newsgroups, so exchanging articles with either will give you the widest possible selection. The bottom line is that it's up to you as the Usenet administrator to choose which newsgroups you want, and you need to make sure that the Usenet host with which your Usenet server will be exchanging newsgroups has the ones that you need. (It's sometimes more important to decide which newsgroups you *don't* need.)

How, exactly, you configure a UNIX-based Usenet server to regularly exchange certain newsgroups with another Usenet server is a topic beyond the scope of this book. These past few pages are intended to provide you with enough background and understanding of how Usenet works to set up and configure the Lotus Notes InterNotes News Gateway; the next section is where you will find a more practical discussion of how Lotus Notes interacts with Usenet.

## THE LOTUS NOTES INTERNOTES NEWS GATEWAY

The Lotus Notes InterNotes News Gateway is a collection of software programs and Notes databases that automate and facilitate the integration of Lotus Notes users and Usenet articles. Running this gateway in your organization means that your users can read and post Usenet articles from their Notes workstations, and it also means that Usenet articles can be included in Workflow and other Notes applications.

### The Players

There are four parts to the Lotus Notes InterNotes News Gateway:

- The Server software
- The News Configuration database
- The Spool database
- The News database

The Lotus Notes InterNotes News Gateway (hereafter, the *News Gateway*) *server software* is a server task that runs continually on a Notes server. This software is responsible for exchanging articles with a Usenet server, posting and organizing articles into a Notes discussion database, and maintaining and optimizing the databases that are used with the News Gateway.

The *News Configuration database* is a Notes database that comes “in the box” when you buy the News Gateway. This database is located on the Notes server that is running the News Gateway software, and is used primarily to determine which newsgroups are exchanged (in other words, *subscribed to*) between the News Gateway and the Usenet server. It also is used to control the frequency of Usenet transfers, as well as a few other and less important configuration settings (such as the name of the Usenet server with which articles are exchanged).

When articles are downloaded by the News Gateway, they are temporarily stored in the *Spool database*. This database is also located on the Notes server that is running the News Gateway. Once the News Gateway has concluded a connection to the Usenet server, it moves the

articles from the Spool database to a discussion database. The Spool database's file name is `INSPPOOL.BOX`. This database is created automatically when you install and configure the News Gateway.

**T I P**

The Spool database is to the News Gateway what the `MAIL.BOX` database is to a Notes server's Router.

The final player (or players) in the News Gateway is (or are) the *News database* (or databases). This is a discussion database that is used by Notes users to read and post Usenet articles. It is also created automatically when you install and configure the News Gateway, and is the database into which the News Gateway moves documents from the Spool database. Also note that, depending on the configuration of the News Gateway and the usage patterns of the users, there can be more than one News database.

**N O T E**

The thing to remember about the News database is that it is, at heart, "just" a Notes database. Although the documents in it are created by Usenet users all over the world, this database can be customized to include special Views, Folders, full-text indices, navigators, and Agents.

## How It Works

As we have seen, the core of the News Gateway is a server task. Much like a Notes server's Router and Replicator tasks, the News Gateway is usually loaded when the Notes server is started, and the program stays in memory as long as the server is "up." See Chapter 26 for a full discussion of how to install and configure the News Gateway.

The chief responsibilities of the News Gateway are to:

- Consult the News Configuration database to determine the name of the Usenet server with which it exchanges articles, and to determine the newsgroups for which articles will be exchanged.
- Initiate connections to a Usenet server.
- Download the names of new newsgroups that have become available on the Usenet server since the last connection.

- Download and upload Usenet articles for the subscribed newsgroups.
- Move recently downloaded articles from the Spool database into the appropriate News database(s).
- Distribute articles from one News database to another.
- Regularly reorganize and compact the News databases, and delete articles that have expired.

Figure 20.3 shows how all these parts fit together.

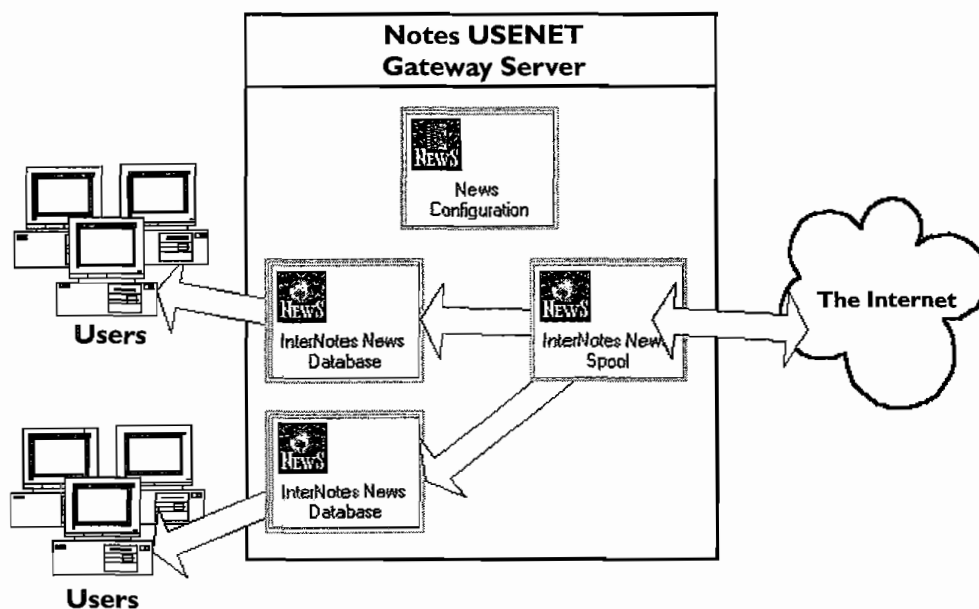


**Figure 20.3** The News Gateway uses the News Configuration database to determine when, where, and how Usenet articles are downloaded.

#### How Articles Get Downloaded, or "Going from Usenet to Notes"

The example below follows a Usenet article as it makes its way from the original Usenet poster anywhere in the world to your Lotus Notes Usenet News Database. (It may be helpful to refer to Figure 20.3 as you read.)

James, who is employed as a financial analyst in Dublin, regularly follows the newsgroup called *biz.currencies*. There's been a raging debate of late concerning bank deregulation, and Patrick has chimed in with his two pence. In other words, he has written a new Usenet article, and he has posted the article to his Usenet server in Ireland.

Through the wonders of TCP/IP, the Internet, and a vast network of interconnected Usenet servers, James' article has made its way to the Usenet server displayed at the top of Figure 20.3. For the sake of our example, this Usenet server is located in Chicago, and is owned by an ISP.

Bill is also a financial analyst. He works in Chicago, and he, too, regularly follows *biz.currencies* in Lotus Notes. James uses the News database called *Our News*, which is located on the News Gateway server pictured in Figure 20.3. This Notes server is named *Marketing\_1*.

The administrator of the News Gateway on *Marketing\_1* has already added *biz.currencies* to the list of subscribed newsgroups. By filling out a News Connection document in the Configuration database, the News Gateway administrator has also already configured the News Gateway for regular connections to the ISP's Usenet server in Chicago. For the sake of our example, the next connection is set for 1:00 P.M..

At 1:00 P.M. the News Gateway calls (via TCP/IP or PPP/SLIP) the Usenet server. Before exchanging new articles, the News Gateway first checks for and downloads any new newsgroups that have become available on the Usenet server since the last connection. Since the News Gateway on *Marketing\_1* connects to the Usenet server every hour, this step doesn't take very long. Any new newsgroups that have been downloaded are automatically added to the News Configuration database.

After the new newsgroups have been taken care of, the News Gateway checks its list of subscribed newsgroups to see if new articles have become available on the Usenet server since the last connection. Since the News Gateway on *Marketing\_1* is subscribed to a bunch of newsgroups in the *alt.\** hierarchy, and since Usenet articles are downloaded in more-or-less alphabetical order based on the name of the newsgroup, it takes a while before the News Gateway gets to the new articles (including James' opinion about bank deregulation) in the *biz.currencies* group. All new articles that have been downloaded are stored in the Spool database.

Once the last article has been downloaded from the Usenet server into the Spool database on *Marketing\_1*, the connection to the Usenet server is closed. Now the News Gateway moves the new articles from the Spool database into the various News databases, as indicated in the Newsgroup documents in the News Configuration database. Depending

on the number of articles that have been downloaded, this step can also take awhile; but the good news is that the News Gateway is no longer connected to the Usenet server, so at least the meter isn't running!

Once the News Gateway has moved the new articles from the Spool database into the News database, it has to reorganize the documents in the News database to assemble the documents into a response hierarchy (unless the `Response Hierarchies` field in the database's News Database document is set to `No`).

Bill, the guy waiting so patiently to read the new articles in *biz.currencies*, will not find the article from James in the News database until the News Gateway has completed the step of moving the articles from the Spool database into the News database; and he will not find James' article arranged with the rest of the articles about deregulation until the News Gateway has completed the step of building the Response Hierarchies.

#### How Articles Get Uploaded, or "Going from Notes to Usenet"

The previous example about James and Bill illustrated how articles that have been posted to the wide, wide, world of Usenet make their way into a News database on a Notes server that's running the News Gateway—but that's only one-half of the story. Usenet is about discourse, and discussion, and debate; so the News Gateway also has to take articles that have been created by Notes users in the News database and distribute them to the millions of Usenet users around the world. Refer to Figure 20.4 as you follow the process.

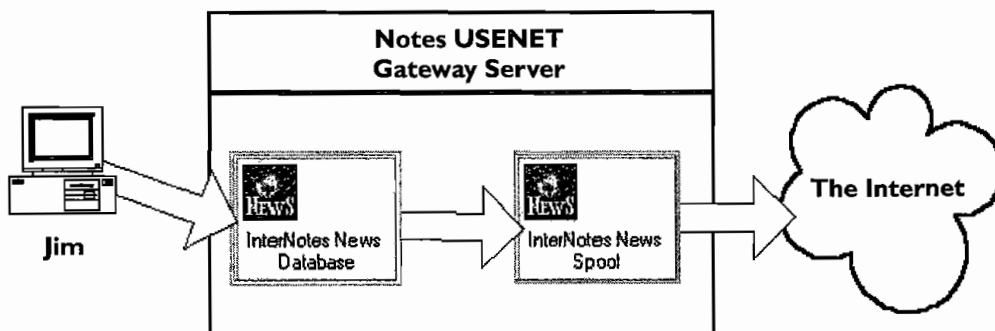


**Figure 20.4** The News Gateway also takes care of uploading of Usenet articles that have been created by Notes users.



To set the stage for this example, imagine that Laurie uses Notes to read Usenet articles, particularly *comp.apps.spreadsheets*. Ann works at same company as Laurie, and she is also a regular follower of *comp.apps.spreadsheets*, but Ann reads *comp.apps.spreadsheets* in a different News database.

Now imagine that Laurie has posted a new article for *comp.apps.spreadsheets* to her News database. Two things have to happen to that new article: (1) it has to get copied from Laurie's News database into the Spool database for eventual upload to the ISP's Usenet server (and distribution to the rest of the world) and (2) during the next connection to the Usenet server, the article has to be delivered to Usenet.

In addition to the regularly connecting to a Usenet server, the News Gateway also scans any and all News databases for new articles. (The frequency with which the News Gateway checks News databases for new articles is controlled, as you'd expect, in the News Configuration database.) Since the default time period for moving new documents into the Spool database is fifteen minutes, it's not long before Laurie's article is moved from her News database. Laurie's article will remain in the Spool database until the News Gateway's next connection to the Usenet server.

When the News Gateway connects to the Usenet server, its first task is to download the new articles on the Usenet server that have arrived there since the last connection. These articles, as we have seen, are temporarily stored in the Spool database. Once the download is complete, it's time to upload, (to deliver any documents that have accumulated in the Spool database since the last connection). This delivery proceeds in an organized and alphabetical manner, so any articles pending delivery for, say, the *alt.\** newsgroups get dispatched before the articles bound for the *biz.\** hierarchy. Once an article is successfully delivered to the Usenet server, it is removed from the Spool database. (And once the article is received by the Usenet server, it becomes the responsibility of that Usenet server to get the document delivered to the Usenet community around the world.)

As we have seen in previous examples, the News Gateway also downloads articles from the Usenet server, and Laurie's article is no exception. Her article is treated just like any new, inbound Usenet article. The new incoming article from *comp.apps.spreadsheets* (which just

happens to be from Laurie) is delivered to the Spool database on the News Gateway server.

The article that Laurie created is not subsequently posted to Laurie's News database, since the News Gateway is smart enough to recognize that the article originated in that database. The article is, however, delivered to Ann's News database. The irony is that Ann won't find Laurie's article in her News database until the article has made the round-trip from Laurie's News database to the News Gateway server, to Usenet, and back again to the News Gateway (but bound for Ann's News database on the return trip).

## SUMMARY

- The Lotus Notes InterNotes News Gateway integrates Usenet with Lotus Notes databases.
- The Gateway consists of the software that runs on the server, and at least three Notes databases: the News Configuration database (where configuration information is stored), the Spool database (where incoming and outgoing articles are stored), and the News database(s) (that users open to read Usenet articles).
- The News Configuration database is used to manage which newsgroups are downloaded from the Usenet server, and how often they are downloaded.
- Articles flow from the News database to the Spool database to Usenet, or from Usenet to the Spool database to the News database(s).



# **INSTALLING AND CONFIGURING THE INTERNOTES NEWS GATEWAY**

Installing the InterNotes News Gateway is a matter of copying software and database templates onto the Notes server, and then filing out a few documents in the databases to configure the server. This is a simple affair—completion time is measured in minutes (as opposed to hours!)

In this chapter, you will learn about: a discussion of the practical hardware & software requirements, the steps involved in bringing up the News Gateway for the first time, and how to subscribe to Usenet newsgroups.

## **HARDWARE AND SOFTWARE REQUIREMENTS**

It's a cruel twist of fate that the hardware and software requirements that are listed "on the box" for most computer products are put there by the company's marketing department. Since these fine folks want you to buy the product, you'll find that the list of what you need to run the product is often, well, *underestimated*. In other words, there's usually a big difference between the hardware that you *can* have to run the product and the hardware that you *should* have to run the product.

Some of this may be obvious, but for the record, you'll need the following to run the Lotus Notes InterNotes News Gateway:

- A Lotus Notes (R4.1 or greater) server
- A network adapter
- A TCP/IP Connection
- A Usenet server, or the ability to connect to one

The manuals also list the minimum CPU and speed as an 80486 with 16Mb of RAM and 10Mb of disk space. Although it is virtually impossible to predict the exact server requirements for the "box," any administrator worth his salt will want a lot more, well, *horsepower*. When determining how much speed and memory you'll require for your News Gateway server, consider such factors as:

- How many users will be connecting to the server?
- What are the other Notes services for which this machine will be responsible? (i.e. is the server also used for some other server task, such as the Web Retriever?)
- How often and for how long this machine will be connecting to the Usenet server?
- How many newsgroups will be retrieved on a regular basis?
- How many articles are posted to the subscribed newsgroups each day?

## INSTALLATION AND CONFIGURATION

### Where Have All the Files Gone?

The following files get installed with the News Gateway:

- INNEWS.EXE, which is the InterNotes News Gateway program
- NNNTTP.DLL, which is the NNTP API code for either Windows or OS/2

- INNEWS.NTF, a database template that is used whenever the Gateway has to create a News database
- INCONFIG.NTF, the template used to create the Gateway's configuration database
- INNEWDOC.NSF, the online documentation

The first two items in this list (the .DLL and .EXE files) are installed in the server's program directory, while the templates and documentation database go in the server's data directory.



To store the InterNotes news or configuration databases in a directory other than (or "below") the Notes server's default data directory, add the line **INDefaultDir=directory name** to the server's NOTES.INI.

Although seemingly insignificant, the location of these configuration databases and more important, the News Gateway's ability to find them, is importantly, so this topic bears a little more explanation. Let's say that the Notes server's program directory is C:. The server's program data directory (as indicated by the `Directory=` statement in the server's NOTES.INI file) is C:. And, for the sake of this example, we decide to put the News Gateway's configuration databases and templates in a directory beneath this data directory.

This scenario requires that the line **INDefaultDir=news** be added to the server's NOTES.INI; the Gateway would not be able to function without it. Notice that the directory name specified in the `INDefaultDir` statement is relative to the server's data directory—the entire C: path is *not* specified.

If you choose to keep the News Gateway's configuration databases and templates right in the Notes server's data directory, do not add the `INDefaultDir` to the NOTES.INI file.

### Making the Server Run the Gateway Task Automatically

In most cases, a Notes server is configured so that it automatically loads the News Gateway software. This, too, requires a modification to the server's NOTES.INI, specifically to add **innews** to the server's task list.

Open the NOTES.INI file for the server with any text editor, and search for the line `ServerTasks=`. At the end of the line, add a comma and the word **innews**. The NOTES.INI should look like the one pictured in Figure 21.1.

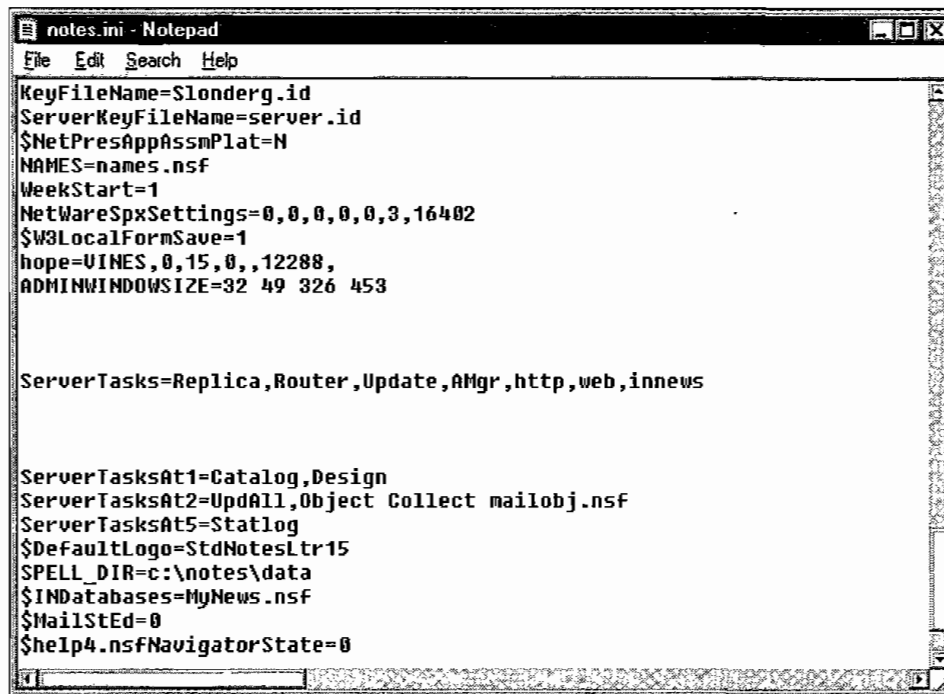
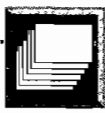


**Figure 21.1** You have to add the **innews** task to your server's NOTES.INI.



**NOTE**

This task can also be added via the server console.

## RUNNING THE GATEWAY FOR THE FIRST TIME

After installing the software, there are seven steps to running the News Gateway for the first time:

1. Loading the **innews** task on the Notes server

2. Creating a News Gateway document in the News Configuration database.
3. Creating a News server document in the News Configuration database.
4. Creating a News connection document in the News Configuration database.
5. Downloading the names of the newsgroups from your Usenet server.
6. Creating a News database document in the News Configuration database.
7. Subscribing to some newsgroups.

### Loading the INNEWS server task

To load the InterNotes News Gateway task on a Notes server, switch to the Notes server console and type **load innews**. (This step is not necessary if the Notes server has been restarted since the NOTES.INI `ServerTasks=` line was modified.) When this server task is loaded for the first time, the News Gateway automatically creates the two databases that are used to configure and manage the News Gateway, namely:

- **INCONFIG.NSF**, also known as the InterNotes News Configuration database, in which the News Gateway will store the names of all the newsgroups that are available on the Usenet news server, as well as configuration information such as the name of the InterNotes News server, the individual newsgroups for which a Subscription document has been created, and the schedule for news connections.
- **INPOOL.BOX**, which is the database that will be used to temporarily store incoming and outgoing news articles. When the News Gateway downloads fresh news articles, they are stored in this database for later delivery to the individual news databases. Similarly, any messages posted by users are also stored in this database, pending the next connection between the News Gateway server and the Usenet News server.

## Creating a News Gateway Document

After successfully loading the **innews** server task for the first time, the Notes server running the InterNotes News Gateway software needs some additional configuration information. To create the News Gateway document that holds this information, open the InterNotes News Configuration database and choose **Create+News Gateway**. See Figure 21.2, which shows that this Form has six fields.

**New Gateway**

News Gateway

Notes Server Name: Marketing 1/Braintree

Description: (Marketing dept's usenet server)

Default SMTP Gateway: Internet

Default Article Expiration: 15 days

Log Level: Verbose

Cross Post and Spool Interval: 15 minutes

This tells the server how much information to post to the screen and the Notes log

**Figure 21.2** Step 1 in configuring your news server is creating a News Gateway document.

- Notes Server Name, which contains the distinguishing name of the server that will be running the InterNotes News Gateway software;
- Description, which may contain an entirely optional comment about the server;
- Default SMTP Gateway, which holds the name of the Internet domain through which e-mail messages are sent to your company or organization. When Notes users post Usenet articles, the articles will be marked as having come from `UserName@Whatever` is in the Default SMTP Gateway field.



- **Default Article Expiration**, which contains the number of days after which the Gateway can remove documents from the News database. (Since Usenet databases have a tendency to grow quite large, consider an expiration period shorter than the default of 15 days.)
- **Log Level**, containing either the word *Verbose* or *Terse*. *Verbose* prints a lot of diagnostic information and messages on the Notes server console while the News Gateway is running; *Terse* lists less. You might set this field to **Verbose** for a few days after installing the News Gateway software, and then set it to **Terse** after you are certain that everything is configured and running smoothly.
- **Cross Post and Spool Interval**, which contains the frequency with which the News Gateway will examine multiple News databases (if any) for duplicate articles. This field also determines the frequency with which the News Gateway examines the News database(s) for new articles that have been created by users and which should be moved to the INSPOOL.BOX database for eventual delivery to the Usenet server.



It is a convention of Usenet that the e-mail address of an article's author be listed in the article, presumably so that a reader anywhere in the world knows how to get in touch with the person. When a Notes workstation user posts an article to Usenet via the InterNotes News Gateway, her e-mail address is listed as `user name@whatever` you enter in the Default SMTP Gateway field. This presumes that the user has access to the SMTP MTA for the delivery and receipt of Internet-based mail messages. Without this Gateway, Usenet readers (worldwide) will not be able to send e-mail to your users.

## Creating a News Server Document

The next step on the road of configuring the News Gateway is to create a News Server document, so that Notes knows the name of the Usenet server with which it will be exchanging Usenet articles. Open the InterNotes News Configuration database, and choose **Create+News Server**. This mercifully short Form has only two fields, as shown in Figure 21.3.

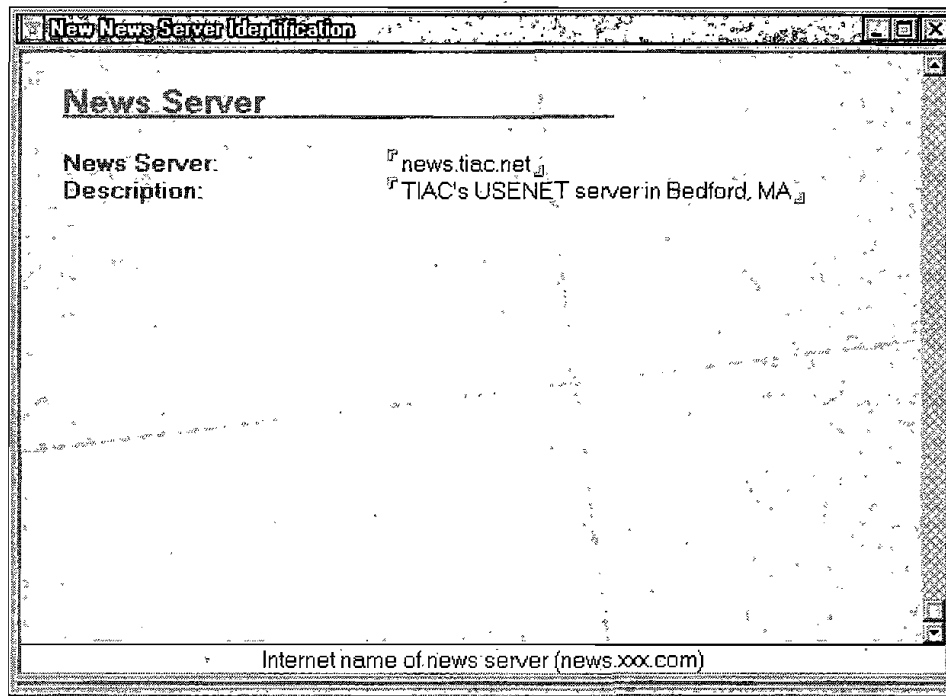


**Figure 21.3** Step 2 in configuring your News server is creating a News server document.

- **News Server**, containing the fully qualified Internet name of the Usenet server with which your InterNotes News Gateway server will be exchanging articles (such as *news.tiac.net*).
- **Description**, which may contain an entirely optional comment about the server.

### Creating a News Connection Document and Scheduling News Transfers

The News Gateway uses a News Connection document to determine how often it will exchange articles with a Usenet server. Open the InterNotes News Configuration database, and choose **Create+News Connection**.

**New Connection**

News Connection

**From the News Gateway:**

**To the News Server:**

Scheduled News Exchange

**Enabled/Disabled:**

**Feed Type:**

**Receive Articles:**

**Post Articles:**

**Poll at times:**

**Repeat interval of:**

**Days of week:**

Comments

Press enter for a valid list of News Servers

**Figure 21.4** Step 3 in configuring a News server is creating a News Connection document.

Figure 21.4 shows the ten fields on this Form. Most of the fields have been set up as keyword choices; press **Enter** to see the list of choices for each of the fields. The fields on the News Connection Form are:

- **From the News Gateway**, containing the full, distinguished name of the Notes server that is running the News Gateway software.
- **To the News Server**, for the name of the Usenet server with which the Notes server will be exchanging articles.
- **Enabled/Disabled**, which is used to turn off or on the News connections. The default value for this field is **Enabled**, but you have the option to occasionally **Disable** the server for maintenance and backup purposes.
- **Feed Type**, for which the choices are **Pull** or **Push**. This field determines if the Notes server will initiate the Usenet connections to the Usenet server, or if the Usenet server will initiate the Usenet connections to the Notes server.

- **Receive Articles, (Yes or No)** The default value for this field is **Yes**.
- **Post Articles, (Yes or No)** Set this field's value to **No** to prevent users from posting articles to Usenet.
- **Poll at Times**, used to store the time for the first connection of the day.
- **Repeat Interval of**, to determine the frequency (in minutes) for news connections between Notes and the Usenet server.
- **Days of Week**, to determine on which days this schedule will be in use.
- **Comments**, containing an entirely optional comment about this connection.

Entering a range of times into the **Poll at Times** field makes the News Gateway connect to the Usenet server between the times entered, and also makes the News Gateway re-connect every **n** minutes, where **n** equals the value in the **Repeat Interval of** field. For example, if you entered **12:00 AM - 10:00 PM** in the **Poll at Times** field, and you entered **60** in the **Repeat Interval of** field, your News Gateway server will begin connecting to the Usenet server every hour on the hour at midnight, and will make the last connection of the day at 10:00 PM. If you entered **12:15 AM - 6:15 PM** in the **Poll at Times** field, and you entered **120** in the **Repeat Interval of** field, the News Gateway server will begin connections for the day at fifteen minutes past midnight, connect to the Usenet server every two hours on the quarter hour throughout the entire day, and make the last connection of the day at 6:15 PM.

Entering a series of times, rather than a range, in the **Poll at Times** field causes the server to connect at certain specific times. For example, entering **1:00 AM, 6:00 AM, 12:00 PM, 6:00 PM** into this field tells the News Gateway server to make four connections per day. Entering a series of time values into the **Poll at times** field makes the News Gateway use the number of minutes in the **Repeat Interval of** field to determine how soon to call back in the case of an unsuccessful connection.



---

Scheduling Usenet connections for a News Gateway server is just like scheduling Routing and Replication events on a "regular" Notes server.

---

## Downloading the Newsgroups Names for the First Time

To subscribe to any of the Usenet newsgroups and begin exchanging articles between the News Gateway server and the Usenet server, it is necessary to first download the names of the newsgroups that are available for subscription onto the Usenet server. With up to fifty *thousand* Usenet news groups in existence, it is a fair assumption that no InterNotes News Gateway in the world will subscribe to all of them! In most cases, the Usenet server with which the InterNotes server will be exchanging Usenet articles probably doesn't have all 50,000 newsgroups, anyway.

Use the server console command **tell innews newsgroups** to instruct the News server to download the newsgroup names from the Usenet server. (This presupposes that a News Gateway document already exists and contains the name of a valid Usenet server.) Downloading these newsgroups names may take some time, depending on (1) the speed of the connection to the Usenet server, (2) the number of newsgroups available on the Usenet server, and (3) whether the News Gateway server is set for **Terse** or **Verbose** logging. The best case for downloading the group names would be a T1 connection to the Usenet server, a relatively small number of groups on the Usenet server, and **Terse** logging. A far worse (i.e. slower) situation would be a 28.8 modem-based connection between the News Gateway server and the Usenet server, a lot (perhaps all 20,000) newsgroups on the Usenet server, and the log value set to **Verbose**. Downloading the newsgroup names can take anywhere from a few minutes up to one or two hours.

The good news is that this large list needs to be downloaded (usually) only once, the very first time the News server is installed and configured. After the first download of newsgroup names, the News Gateway server will check for new groups on the Usenet server, but the number of new newsgroups created on the Internet each day is very small—on the order of three or four.

The next step in configuring the News Gateway server is to choose the groups with which the InterNotes News Gateway server and Usenet server will actually exchange articles—but this can't be done until the entire list of available groups has been downloaded from the Usenet server. Use the server console command **show tasks** to check the progress of the download.

## CREATING A NEWS DATABASE

The News database is the Notes database that Notes users open to read Usenet articles. As we have seen, the **innews** server task downloads articles from the Usenet server, and temporarily saves them in the InterNotes News Spool database. Once a newsfeed has completed, the **innews** server task then moves the new articles from the spool database into the News database(s).

Although more than one news database can be established, the fewer, the better. A common mistake made by the newbie News Gateway administrator is to create a separate news database for each subscribed Usenet group; nascent administrators also often make the mistake of creating a separate news database for each of their users. As we've seen, the news database(s) can grow quite large, so it stands to reason that the fewer of these databases, the better.

Another reason to limit the number of News databases has to do with the way the News Gateway processes incoming Usenet articles. Notes users are familiar with the threaded appearance of a discussion database, wherein response documents are listed in indented form, underneath their parent documents. This visual representation of articles and responses is used in the News Gateway's News database, too, as seen in Figure 21.5.

This facility for a response hierarchy is built-in to Notes documents that are created by Notes users. In other words, when Notes users create response documents in Notes databases, Notes automatically keeps track of the documents' parents; and based on this association, Notes is able to maintain and display the aforementioned response hierarchies for those documents. For the record, this response hierarchy is stored in a special field called `$Ref` that is automatically created by Notes for each and every response document.

Now consider a discussion thread of, say, five related news articles from Usenet. The articles themselves have been created by five different people on five different kinds of computers in five different parts of the world. When the News Gateway server collects these five articles from the Usenet server, Notes has no way of knowing that these five articles are all related to one another; as we've seen, Notes response hierarchies are based on `$Ref` fields. And since these articles were not originally created in Notes, there's don't have `$Ref` fields. And since there's no `$Ref` fields, there's no response hierarchy.

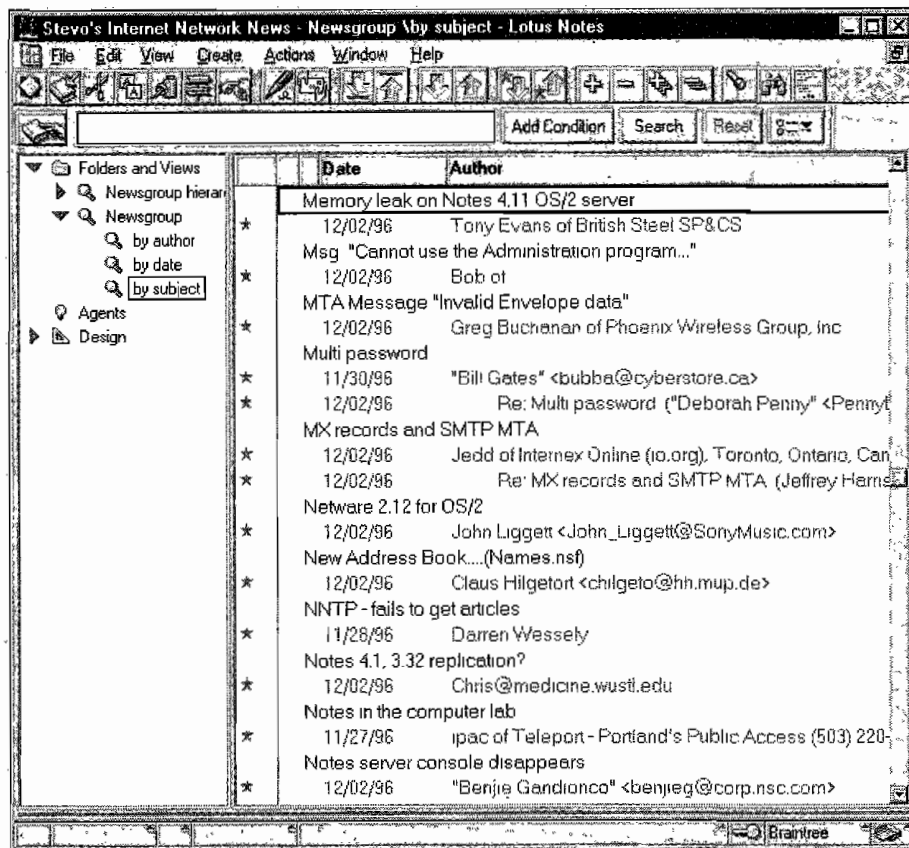


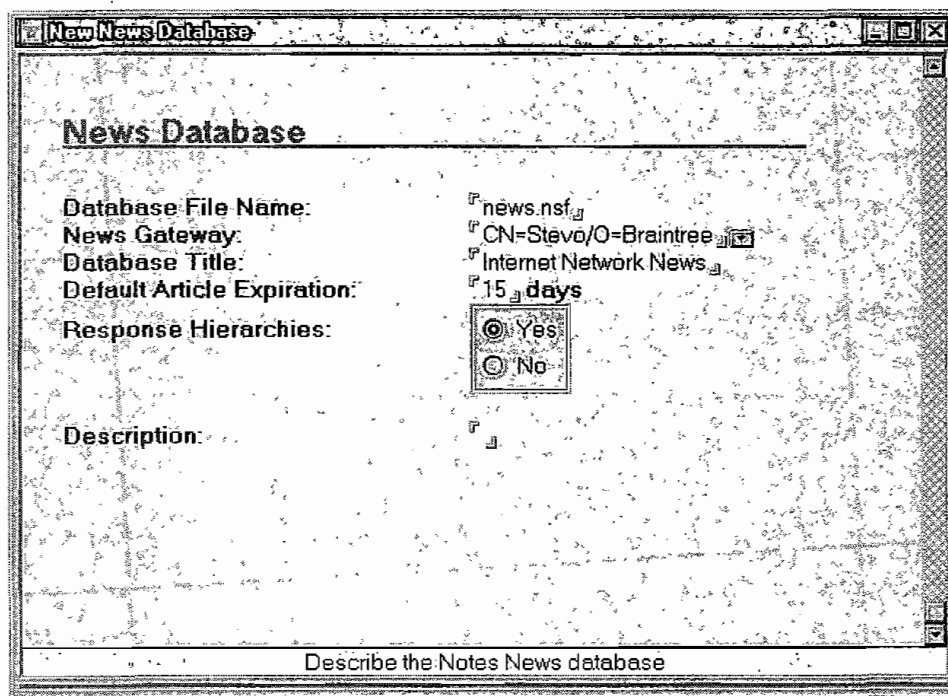
**Figure 21.5** The News database automatically threads the articles in a response hierarchy.

So, this, then, is the final undertaking for the News Gateway: after moving new articles from the Spool database to the News database, the News Gateway does its best to create the \$Ref fields needed by Notes to display the five articles in a response hierarchy. The News Gateway analyzes the incoming documents, and groups them together based on the contents of the Usenet articles' Subject fields.

This can be a time-consuming and CPU-intensive proposition, especially if the server receives articles from a lot of high-traffic Usenet groups.

**The bottom line:** It's a lot of work for the News Gateway to create response hierarchies for related Usenet articles. To conserve processing cycles, use as few News databases as possible. If you have seven users who want to subscribe to *alt.sport.paintball*, don't create seven separate news databases. It's better to have the seven people all use the same News database.

To configure the News Gateway for a News database, open the InterNotes News Configuration database, and choose **Create+News Server**. There are six fields on the News Database Form, as you can see in Figure 21.6.



The screenshot shows a window titled "New News Database". Inside, there's a section titled "News Database". Below this, there are several fields and controls:

- Database File Name:** A text field containing "news.nsf".
- News Gateway:** A text field containing "CN=Steve/O=Braintree".
- Database Title:** A text field containing "Internet Network News".
- Default Article Expiration:** A text field containing "15 days".
- Response Hierarchies:** A group box containing two radio buttons: "Yes" (which is selected) and "No".
- Description:** A large text area containing the text "Describe the Notes News database".

**Figure 21.6** Creating a News Database document.

The six fields on the News Database Form are:

- **Database File Name**, containing the file name for the Notes database that will hold Usenet articles;
- **News Gateway**, containing the name of the Notes server that is running the **innews** News Gateway server task. This is a keyword field; press **Enter** to select the server's name from a list;
- **Database Title**, holding the name for the Notes database that is going to hold Usenet articles. The database Title is the name by which the database will be known in the **File+Database+Open** dialog box, and is also the name that is displayed in the icon for this database on a user's desktop.



- **Default Expiration**, containing the number of days after which the News Gateway can delete articles from the database. (This setting overrides the `Default Expiration` field in the News Gateway Form.) If you have several News databases that need to have different expiration settings, use this field; otherwise, leave it set to the default value of 15 days.
- **Response Hierarchy**, containing either **Yes** or **No** and used to determine whether or not the News Gateway should create the `$Ref` fields that are necessary for the display of indented responses in the database's views. Setting this field to **No** can considerably increase the speed and performance of the Gateway and for the delivery of new articles—at the expense of the visually-appealing and informative response hierarchies that make it easy for users to follow discussion threads in the News database.
- **Description**, containing an optional comment about this database.

When a new News Database document is saved, the News Gateway will automatically create a new Notes database. This design of this new database is based on the `INCONFIG.NTF` template, and the database's Title and file name will be whatever is contained in the corresponding fields in the News Database Form.

**NOTE**

The News Gateway checks for work to do approximately once a minute, so you may have to wait all of sixty seconds before opening the new News database.

## SUBSCRIBING

When the News server downloads the available newsgroup names from the Usenet server, the News Gateway creates a document for each in the News Configuration database. Subscribing to a group—in other words, telling the News Gateway to regularly download the articles from a particular newsgroup—is a matter of opening the Configuration database and running an agent in the database “against” the desired newsgroup document(s).

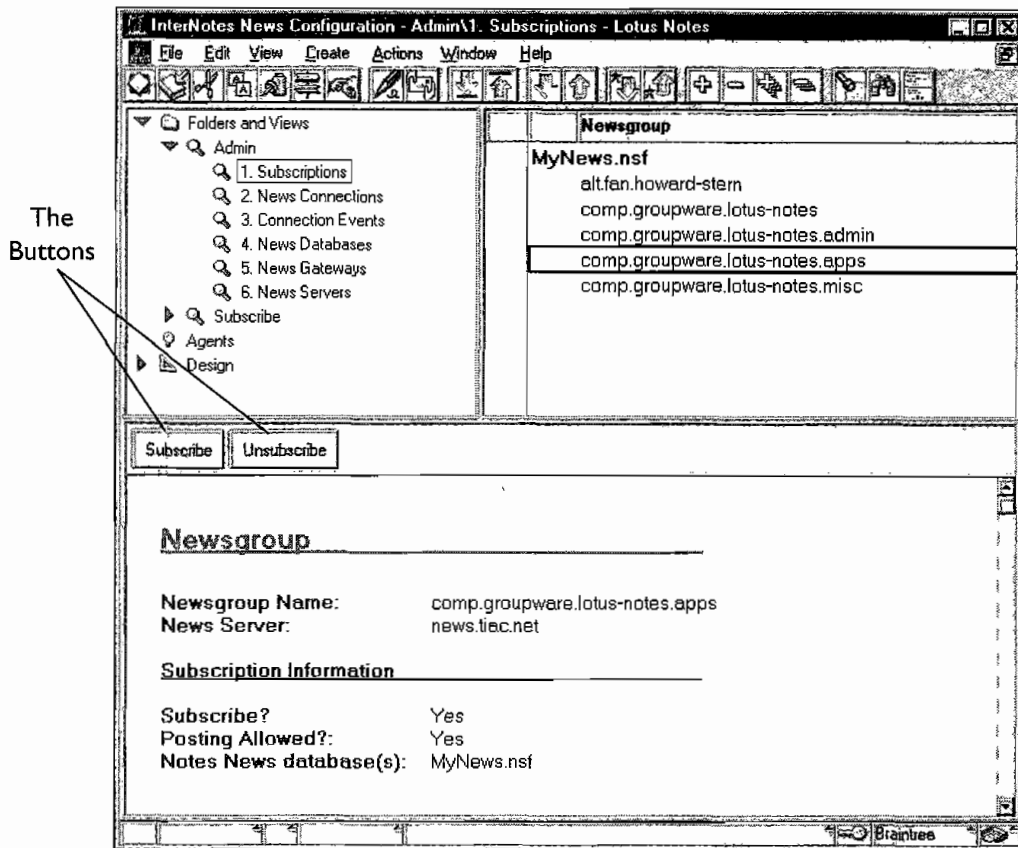


**Figure 21.7** Use the buttons in the Admin Subscriptions view to determine which newsgroups are exchanged with the Usenet server.

To subscribe to one or several newsgroups, open the **SubscribeNewsgroup** View in the InterNotes News Configuration database, select the document(s) corresponding to the desired newsgroup and choose **Actions+Subscribe Macro**. Notes will prompt for the name of the news database that will be used to store articles for the selected group(s). Subscribed groups are marked by an icon in the database's View; it is also possible to confirm subscriptions by using the Admin Subscriptions view in the Configuration database.

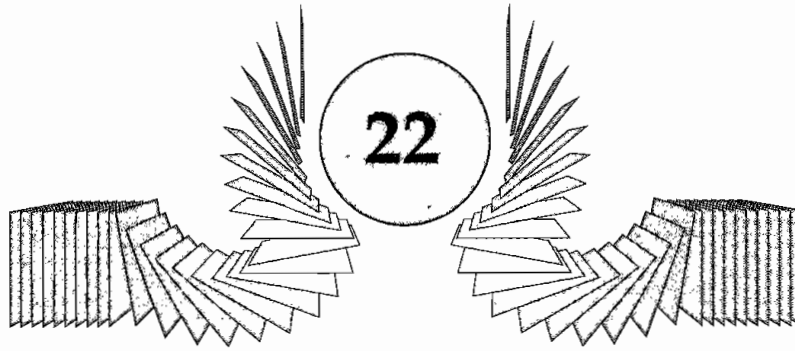
Once a group document has been marked for subscription in the News Configuration database, the News Gateway will automatically download articles for that group and post them in the News database. The first download for the newly-subscribed-to group will take place during the next scheduled connection, as defined in the Server

Connection document. Use the server console command **tell innews transfer** to initiate an immediate download (in other words, without waiting for the next scheduled connection).

Fair warning: the first-ever download of articles may be a prolonged affair, depending on the number of subscribed groups, the speed of the connection to the Usenet server, and on the number of articles in the newsgroup(s).

## SUMMARY

- The InterNotes News Gateway is run on a server by adding the **innews** task to server's NOTES.INI.
- The News Configuration database is used to determine such configuration options as the name of the Usenet server, the frequency of downloads, and the name of the database that users will open to read Usenet articles.
- Any Usenet Newsgroup can be subscribed to by using the **Actions+Subscribe Macro** agent in the News Configuration database.



# **USING AND MANAGING THE DOMINO NEWS GATEWAY**

Any time you add a new task to a Notes server, the first few weeks tend to require the most attention. The good news is that after the dust has settled on a new News Gateway installation, the administrator's only day-to-day task is to process subscription requests from users. The humdrum administrative work, such as removing old news articles, managing disk space taken up by the News database(s), and making discussion threads appear together in Views, is handled, automatically, by the News Gateway. In this chapter, we'll discuss the day-to-day management of the News Gateway, and we'll review how it all works.

## **THE DOMINO NEWS CONFIGURATION DATABASE**

Almost all of the day-to-day administrative tasks for the News Gateway are managed via the Domino News Configuration database. This is the place to go when you need to change the list of newsgroups to which you subscribe, change the frequency with which you exchange articles with the Usenet server, or see a log of previous connection events.

## The Configuration Database's Views

The News Configuration database has eight Views, which you use to control the operation of the Gateway. The Views are:

- Admin\Subscriptions
- Admin\News Connections
- Admin\Connection Events
- Admin\News Databases
- Admin\News Gateways
- Admin\News Servers
- Subscribe\by Newsgroup
- Subscribe\by Newsgroup Hierarchy

All of the Admin Views serve a similar purpose; these Views display the various configuration options for the News Gateway, while the Subscribe Views list all of the Usenet newsgroups that are available on the Usenet server. You use the Admin Views to control the various aspects of the News Gateway's configuration and operation, and you use the Subscribe Views to control which newsgroups are marked for periodic download (in other words, which are marked as *subscribed*).

### Managing Subscriptions

The Admin\Subscriptions View lists all of the newsgroups that are subscribed to by the News Gateway, sorted by the News database into which the news articles are delivered. This view can be particularly helpful when you need to reduce the amount of download time and disk space used by the News Gateway, since it presents a "one stop shopping" view of the various news databases that have been set up, as well as the newsgroups to which the databases are subscribed.

As we saw in Chapter 22, downloading the same newsgroup into many news databases represents a lot of work for the News Gateway, particularly when it comes to creating the Response Hierarchies for the database Views. If you find that you have many users who are subscribing to the same newsgroup(s), consider creating a single, centralized database with these newsgroups, and then "point" your

users at this database. If it takes 20 minutes for the News Gateway to create Response Hierarchies after each download of a particular newsgroup, and you have that newsgroup being downloaded into six different News databases... well, you get the idea. You'll also see a big savings in disk space usage, as well as overall server performance.



The good news is that, when you do have multiple news databases capturing the same newsgroup, the articles themselves are only downloaded once from the Usenet server. As we have discussed, new articles are stored temporarily in the News Spool database when the News Gateway downloads from the Usenet server. When the News Gateway finishes downloading all of the articles for a given connection, it then delivers the articles by moving them from the Spool database into the individual News database(s).

By definition, all of the newsgroup listed in the Admin\Subscriptions View are marked for regular download from the Usenet server. If you decide to unsubscribe from a particular newsgroup, select the newsgroup in the Navigation pane, and choose **Actions+Unsubscribe Macro**.

If you need to change the way in which you subscribe to a particular newsgroup, select the newsgroup in question and choose **Actions+Edit Document** to edit the document. If you want to consolidate the subscriptions for a particular newsgroup into one database, edit the newsgroup document and remove all but one of the database names from the Notes News database(s) field. Refer to Chapter 23 for a full explanation of each of the fields in the newsgroup document's form.

### Managing Connections

The Admin\News Connections view lists all of the connections that are scheduled between the News Gateway and the Usenet server. In most cases, you will have only one connection document in the News Connections view; if you need to change anything about the connection, select the Connection Document and choose **Actions+Edit Document** to edit the document. Refer to Chapter 23 for a blow-by-blow description of each field on the form.

It is conceivable to have more than one news Connection Document; a common scenario is to set up a schedule for regular, hourly connections Monday through Friday, and less-frequent, perhaps twice-daily connections on Saturdays and Sundays. You can also use multiple

Connection Documents if you need to schedule “around” other server tasks; if you know that your News Gateway server is busy on Fridays (it’s backup day!) you could have two Connections Documents to allow for a special connection schedule on that one day of the week. To create a new News Connection Document, choose **Create+News Connection**. Refer to Figure 22.1.

**New Connection**

News Connection

**From the News Gateway:** CN=Stevco/O=Braintree

**To the News Server:** news.tiac.net

**Scheduled News Exchange**

**Enabled/Disabled:** Schedule Enabled

**Feed Type:** Push

**Receive Articles:** Yes

**Post Articles:** Yes

**Poll at times:** 12:00 AM each day

**Repeat interval of:** 60 minutes

**Days of week:** Mon, Tue, Wed, Thu, Fri, Sat, Sun

**Comments**

**Figure 22.1** Connections between the News Gateway and the Usenet server are controlled with News Connection documents in the News Configuration database.

### Viewing the Connection Log

The Admin\Connection Events View presents a log of previous connections; use this View to verify that the News Gateway is connecting to the Usenet server according to the schedule you specified with the News Connection document. The News Gateway automatically creates a document for this view each and every time it connects, or tries to connect, to the Usenet server, so this is the first place you’d look if you suspected something was amiss with your connection schedule.



Each time the News Gateway connects to the Usenet server, it requests articles and new newsgroup names that have been created since the last connection. If, for some reason, you need to re-download all of the articles for all of the newsgroups to which you have subscribed, you can delete all of the documents in the Connection Events View in the News Configuration database. Upon the next connection, the News Gateway will re-download all of the articles for your subscribed newsgroups.

### Managing News Databases

The Admin\News Databases View lists all of the News databases that are being regularly updated by the News Gateway. Use this View to manage the News databases you have created, and also to fine tune what happens, exactly, when articles are posted to a particular News database. To edit a News database, select the document and choose **Actions+Edit Document**. The most common reason for fine-tuning a News database is to change the Default Article Expiration. This field determines how often the News Gateway will delete the articles in this database. The consequences of having a too-long expiration are probably obvious; the longer you save articles, the larger the database will grow. The exact setting for this field depends very much on the way your users read Usenet articles; the basic rule of thumb is that the shorter the expiration period, the better.

You can also edit the News database document to turn on or off the Response Hierarchy. When this feature is turned off, the News Gateway will not go to the trouble of creating the \$Ref fields that are needed for discussion-oriented newsgroups (and which make the documents indent underneath themselves in the database's Views). Since creating these response hierarchies is a time-consuming and CPU-intensive operation, look for any and all opportunities to turn it off. If you are subscribing to a non-discussion oriented newsgroup such as *rec.skiing.announce*, you might be able to get away without building the Response Hierarchies. If, on the other hand, you subscribe to *alt.agriculture.fruit*, you'd probably want to leave this field turned on.



Before you decide to turn off Response Hierarchies, remember that this setting controls all of the newsgroups in the database.



If you need to create a new News database, choose **Create+News Database**. When you finish filling out the fields on the form and save the document, Notes will automatically create the new News database for you.

#### Managing News Gateways

The Admin\News Gateways View in the News Configuration database lists all of the Notes servers at your company that run the News Gateway software. You can edit the News Gateway document(s) to change some of the default configuration settings for the News Gateway; to do so, select the document and choose **Actions+Edit Document**. The most common reason for fine-tuning a News Gateway is to change the `Default Article Expiration`. Whenever you create a new News Database document, the Default Expiration period for that new database defaults to whatever you enter in this field on the News Gateway form. Since having a shorter expiration period can save you lots of disk space and processing time, you might consider lowering it.

Refer to Chapter 24 for a full explanation of each of the fields on the News Gateway form.

#### Managing News Servers

The Admin\News Servers View lists all of the Usenet servers with which your News Gateway regularly exchanges Usenet articles. This View is used as a lookup by some of the other documents in the News Configuration database; for example, when you create a News Connection document, Notes uses this view to determine the name(s) of the Usenet servers with which you can schedule a connection.

In the rather unlikely event that you need to connect to more than one Usenet server, choose **Create+News Server**. It is conceivable to set up your News Gateway to connect to more than one Usenet server. Perhaps you can connect for free to your company's own Usenet server for the bulk of the newsgroups that you need to download, but there are certain groups that are not available on your company's Usenet server. If that is the case, you need another News Server document (as well as another set of News Connection documents) to facilitate the download of articles from some other Usenet server, such as one provided by an ISP.

## More on Managing Subscriptions

Although the various Admin Views in the Configuration database are available to help you manage the various aspects of the News Gateway's configuration, you will probably spend most of your administration time for the News Gateway managing the newsgroups to which you (and, of course, your users) subscribe. When you set up your News Gateway for the first time, you may remember typing the command **tell innews newsgroups** at the Notes server console. This command tells the News Gateway to make an immediate connection to the Usenet server, and to download the entire list of newsgroups that are available for subscription. The upshot of this activity is that the News Configuration database contains one document for each of the available newsgroups.

### Subscribing

Open either the `Subscribe\by Newsgroup` or the `Subscribe\by Newsgroup Hierarchy View` (these two Views list the same information, just sorted differently). Select the document—or documents—for the newsgroup—or newsgroups—to which you want to subscribe, and choose **Actions+Subscribe Macro**. You can also open any one of the newsgroup's documents and manually change the `Subscribe` field to **Yes**. If you manually edit a document to subscribe to that newsgroup, you also have to enter the name of the news database to which you want the articles posted in the `Notes News Database(s)` field.

During the next scheduled connection to the Usenet server, the News Gateway will gather all news articles for the newly subscribed newsgroup, and post them to the aforementioned News database. If you can not wait for the next scheduled connection, you can always type the command **tell innews transfer** at the Notes server console to initiate an immediate connection and download.

There's one other field on the Newsgroup form. Use the `Posting Allowed?` field to decide whether or not your users can post articles to this newsgroup. When you set this field to **Yes**, any documents that your users create in the News Database will be temporarily stored in the Spool database, and then delivered to the wide, wide world of Usenet during the next scheduled connection to the Usenet server. If

you select **No**, your users will only be able to read the articles for this newsgroup; they will not be able to post.

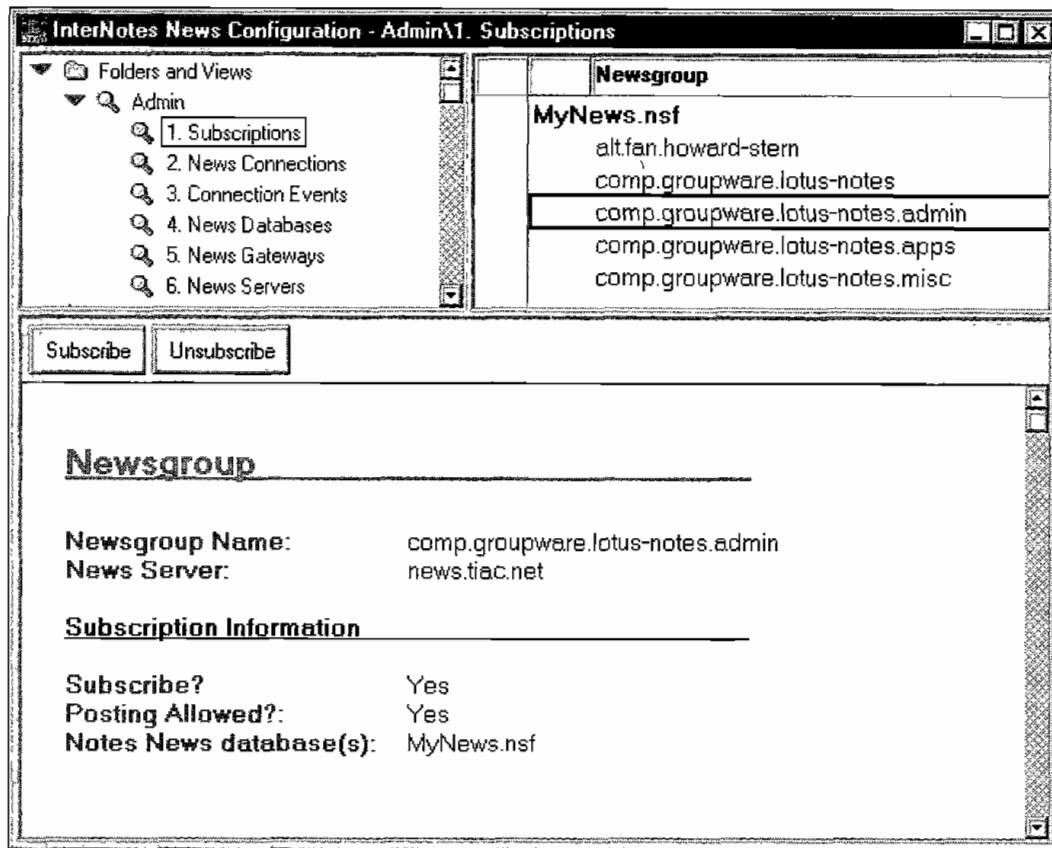


**Figure 22.2** A Newsgroup document in the News Configuration database..

### Unsubscribing

Unsubscribing to a newsgroup is just as easy as subscribing. Open the Admin\Subscriptions View, select the document—or documents—for the newsgroup—or newsgroups—from which you want to unsubscribe, and choose **Actions+Unsubscribe Macro**. Running this macro means that articles from the newsgroup you have selected will no longer be downloaded when the News Gateway connects to the Usenet server.

You might also consider opening the News database that had been the erstwhile recipient of the newsgroup, selecting all of the documents for that newsgroup, and pressing **Del** to delete them.

## DATABASE MANAGEMENT ISSUES

There are just two issues regarding the management of the News Gateway databases: who can open/edit/read the News Configuration database, and how can you manage the news databases themselves?

### Security Considerations for the Configuration database

Undoubtedly, as the News Gateway administrator, you should assign yourself Manager access to the Configuration database. But what about the your users? It's hard to imagine a scenario in which you would want to allow the "rank and file" user to be able to edit such configuration options as the name of the Usenet server, frequency of article downloads, and so on. At first blush, this would seem to indicate that the default level of access for the News Configuration database be set to **Reader**, or perhaps even **No Access**. But at the same time, you have to consider how you want to handle new subscriptions. If you are administering the News Gateway for a large user community, you will almost certainly *not* be able to predict which groups your users want to follow—and that poses an interesting security dilemma.

If you set the default level of access on the News Configuration database to **Reader** or below, users will not be able to subscribe to newsgroups themselves; all such requests will be processed by you. This offers you a great deal of control, since you will have the power to decide whether or not to subscribe to a particular newsgroup. Using **Reader** as the default level of access for this database means that you also will have total control over the size of the news databases and the length of time that your News Gateway connects to the Usenet server—the idea being that if you're already connecting to the Usenet server for too long each time you transfer articles, you'll have the power to veto subscription requests for new newsgroups. (The more groups you subscribe to, the longer each Usenet connection, of course.)

What's the downside to this approach? The downside is that all subscription requests have to be processed by you. Any time a user decides that a subscription is needed for, say, *rec.arts.marching.band.high-school*, you, and only you, will be able to run the Agent in the News Configuration database that adds the newsgroup to the list of articles that the News Gateway downloads. If the usage patterns of your user

community are such that there are lots and lots of groups added and removed from the subscription list, you'll find that you're spending lots and lots of time taking care of them.

If being the single point of contact for subscription requests doesn't turn you on, set the default level of access for the News Configuration database to **Editor**. (Anyone who is going to run the Subscribe/Unsubscribe Agents needs at least Editor access to the News Configuration database.) Granting a user—or all of the users—Editor access to the News Configuration database means that they can subscribe (or unsubscribe) to any newsgroup anytime they want, without having to bother you. No more having to wait on you to process subscriptions.

Needless to say, there's a downside to this approach, too. As we discussed, granting anyone access higher than Reader to the News Configuration database means that they could, if they were so inclined, modify the configuration documents in the database that are used to schedule Usenet connections, in addition to being able to modify the subscription documents. This also means that you are granting users free license to subscribe to as many newsgroups as they want, without care or concern about the size of the News database(s).

It all boils down to what level of access you are going to give users to the News Configuration database. If you need tight control over exactly which groups are marked for download, and over the duration of Usenet connections, stick with Reader. If you can tolerate a more relaxed security model, and if you aren't as concerned about the length of the Usenet downloads, give people Editor access to the News Configuration database. Don't forget, the Reader "model" implies that you are willing to process any and all subscription requests, and that the Editor model absolves you of this responsibility—at the expense of some control over subscriptions and connections.



There another security consideration that's purposely evaded in these pages. Usenet newsgroups range from the culturally enlightening, such as *soc.culture.irish*, to the timely and relevant, such as *biz.marketplace.discussion*, to the salacious and steamy, which begin with *alt.sex*, and then deteriorate quite rapidly in both title and content. Recommendations concerning the propriety of the latter type of newsgroup in the workplace are well beyond the scope and responsibility of this fine technical tome. Having said that, it would be remiss to omit a reminder that allowing any user to subscribe to any newsgroup may result in articles from *alt.sex* (and worse) circulating in your news database.

## Managing the News Configuration and News Databases

The three final, short points to consider about the News databases are related to conserving disk space, creating full-text indices, and how to create a cool Agent.

### Disk space Considerations

The size of the News database is a function of the number of newsgroups to which you subscribe, the daily message traffic in those newsgroups, and the expiration period for the database.

Obviously, some newsgroups produce a lot more traffic than others, and this should be a prime consideration when you are deciding what the database's expiration period should be. Since certain newsgroups routinely exceed one hundred messages per day, most administrators would agree that such newsgroups are best managed with a relatively short expiration period—perhaps even as short as one day. If you find your news databases growing to 100 Mb or more, consider lowering the expiration period for that database, via the News Database document in the News Configuration database. The shorter the expiration period, the more often documents will be deleted from the database, and the less disk space the database will occupy.

As any savvy Notes administrator is aware, deleting a document from a Notes database does not immediately free up disk space; rather, the space *in the database* is marked as available, so that the next new document doesn't make the database any larger. Documents that have reached their expiration date are automatically removed from the News database(s) each night at 2:00 AM. Once the expired documents have been deleted from the database, the News Gateway also automatically compacts the database to remove any whitespace.



One additional note regarding disk space concerns the News Gateway's Spool database; since this database is the temporary holding place for all incoming and outbound articles, it tends to have a lot of whitespace. Since the News Gateway does not automatically compact this database each night, it might be worth your time to schedule (via the Notes server's Program documents) this database for regular compacting.

### Searching

One of the most powerful features of Notes is its ability to do full text searches, and this functionality can be especially germane to a Usenet News database. Let's say you're involved in a marketing project for your company, and one of your tasks is to search Usenet for references to your competition. Subscribing to the right newsgroups is only half the battle; assuming you were in the automotive industry, you might subscribe to seven or eight newsgroups. If those seven or eight newsgroups each produce, say, fifty articles a day, that's about 400 documents to wade through *per day* when you're trying to find mentions of your competitor's name. Full text search is the solution to that problem. When you perform a full-text search on a database, Notes will find for you any occurrence of a word (or string of words) anywhere in the document. This saves you considerable time and effort in a scenario such as our market research example.

Before being able to search a database in this manner, you must create a full-text index for it. To do so, select the News database in question, and choose **File+Database+Properties**, and then click the **Full Text** tab. Clicking the **Create Index...** button creates the index.

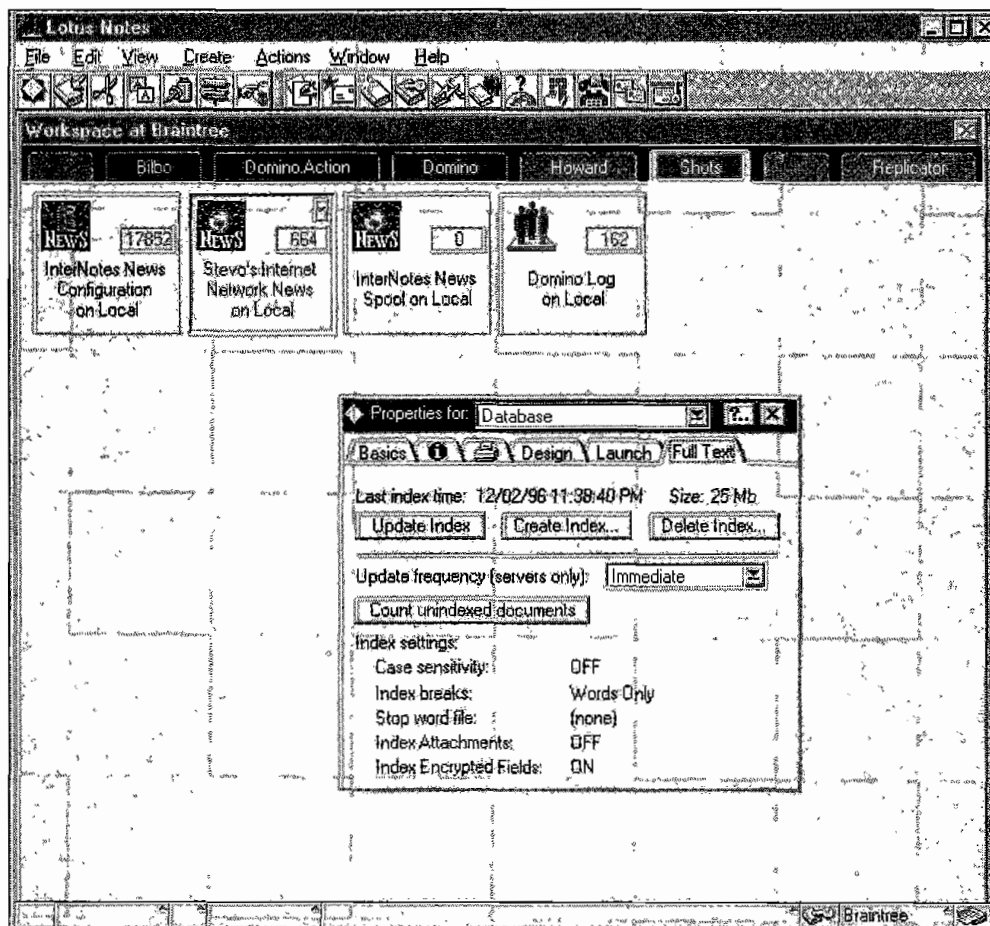


**Figure 22.3** Full-text indices can dramatically increase the usefulness of a Usenet News database.

The only argument against creating a full-text index is that the server has to work to maintain the index as new documents are added and expired documents are deleted from the database. If your server is already at peak capacity in terms of CPU load and/or disk space, creating a full-text index might be beyond its capabilities. Refer to Lotus' Notes Administration Help database for more information about creating and managing full-text indices.



## Search Macros

**Figure 22.4** Search Agents can be used to notify you automatically when certain documents have arrived in the News database.

You can extend the usefulness of a full-text indexed database even further by using a Newsletter Agent. This Agent will, on a periodic basis, search the News database for the criteria you establish, and automatically send you (or anyone else) an e-mail message when documents are found. This frees you from having to open and manually search the database every day, since you are automatically notified when new articles arrive that match your criteria. Figure 22.5 shows an Agent that will search the News database automatically for new documents that include the words *Ford*, *Chrysler*, or *Edsel*.



You must create a full text index before you can create a newsletter agent.

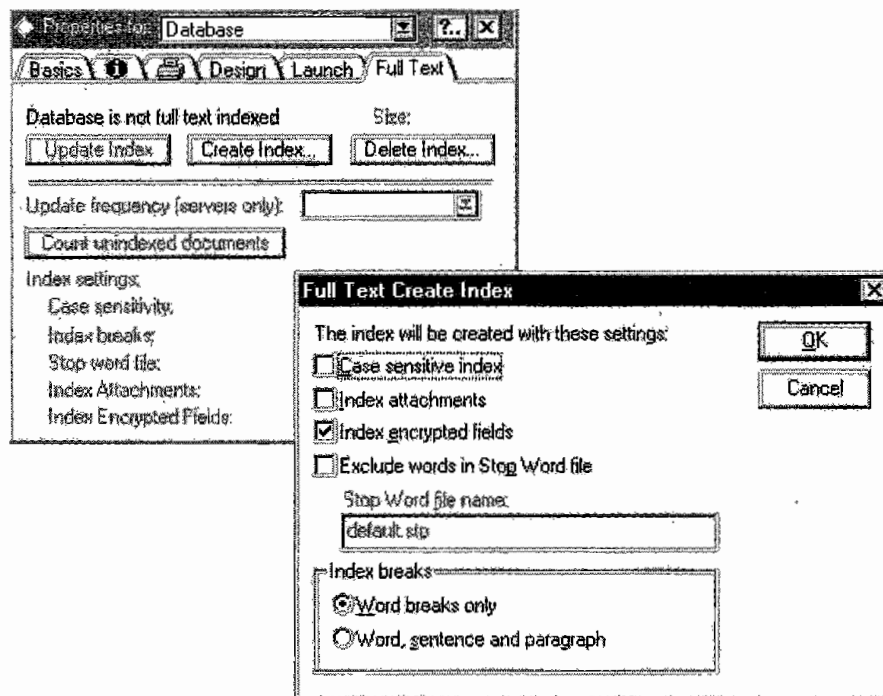


Figure 22.5 Creating an Agent that automatically searches.

## YOUR DAY-TO-DAY WORK

As we have seen, management of the News Configuration and News databases includes both security and capacity issues. Being able to manage and troubleshoot potential problems with the News Gateway also requires a detailed understanding of the Gateway's operations, so it seems about time for a blow-by-blow description of....

### What Happens During Connections to the Usenet Server

Quite a lot transpires during a connection between the News Gateway and the Usenet server. The exact tasks, and the order in which they are performed, are as follows:

1. **Get any new Usenet newsgroups**—Just about anybody in the world can create a new Usenet newsgroup, which may explain the origin (but certainly not the continued existence of) such newsgroups as *alt.chips.salt-n-vinega*. Every time the News Gateway connects to the Usenet server, the first order of business is to download any new newsgroups that have come into being since the last connection. These new newsgroups are automatically added to the News Configuration database, where they can then be marked for subscription. (Or, in the case of our *alt.chips.salt-n-vinega* example, ignored entirely.) You may remember that this step took quite a long time when the News Gateway connected for the very first time, but the day-to-day number of new newsgroups is much, much smaller.
2. **Download new, incoming articles**—The next task is to download all of the new articles since the last connection, but only for the newsgroups that have been subscribed to in the News Configuration database. If this is the first-ever connection for the News Gateway, only the articles that are younger than the expiration period will be downloaded. The new articles are placed into the Spool database to await later distribution to the News database(s).
3. **Upload new, outgoing articles**—Next the News Gateway looks in the Spool database for any articles that have been posted “on your end” since the last connection. If there are any to post, the News Gateway delivers them to the Usenet server. Once the last new, outgoing article has been posted to the Usenet server, the connection to the Usenet server is closed. (If you are paying for the connection to the Usenet server, the meter is running right up until the last outgoing article gets delivered.)
4. **Delivers articles to Notes News databases**—Upon disconnecting from the Usenet server, the News Gateway moves the new articles that it just got from the Usenet server from the Spool database into the News database(s).
5. **Rebuilds Response Hierarchies**—The last job for the News Gateway is to move new documents into Response Hierarchies under existing, related documents (if any).

Step #2, where the News Gateway downloads new articles, takes the longest. Exactly how long depends on both the number of newsgroups that you have subscribed to, and the number of new documents that have been posted to the subscribed newsgroups.

## What Happens at Night

2:00 A.M. is a special time for the News Gateway. While you're home sleeping, the News Gateway has some housecleaning to take care of. Note that other, more general Notes server maintenance is also, by default, scheduled for 2:00 A. M., so you may want to reschedule those Notes server tasks. You can not change the time of the News Gateway's nightly chores. Here's what happens:

1. **Delete Expired Articles**—The News Gateway peruses the News database(s), and removes any articles that are older than the database's expiration period. Note that if the main topic document of a particular discussion thread expires before the response documents', the Main Topic is left alone until all of the Response documents have exceeded their expiration period. If a new Response document is downloaded whose Main Topic has already been deleted, the new response becomes a Main Topic document.
2. **Compacts the News database(s)**—The last task for the wee small hours of the morning is to compact the News database(s) and recover any disk space from the articles that were just deleted.



### NOTE

Compacting the News database is no different from compacting any other database—there must be enough disk space available on the server to make a copy of the entire database file.

## Regular Tasks

In addition to the nightly tasks and the things that happen during a Usenet connection, the News Gateway has the following work to do, approximately every five minutes:

1. **Synchronizing News databases**—The News Gateway scans any and all News databases and moves new documents amongst them, as appropriate. For example, if you have five users who subscribe to *biz.ergonomic\_sciences*, and one of them posts a new article in his News database, the News Gateway will post that new article in each of the other user's News database. (This constant synchronization of databases can create quite a lot of network traffic and server activity if you have active and prolific users. This is yet another argument for having as few News databases as possible.)
2. **Moving new postings to the Spool database**—After your users' new articles have been posted to all of the other News databases that subscribe to the same newsgroup, the News Gateway posts the same article to the Spool database for delivery to the Usenet server during the next connection.

## SUMMARY

- The Configuration database is used to control such options as how often the News Gateway connects to the Usenet server and which groups are downloaded.
- You should think carefully about the access level you will grant users to the News Configuration database.
- Housecleaning is performed by the News Gateway every night at 2:00 A.M.
- Keeping the expiration period for Usenet articles short means that the News Gateway will have less work to do, and also that there will be more disk space available on the Notes server.

# SECTION



# NOTES TO NOTES



## **CONNECTING NOTES TO NOTES VIA THE INTERNET**

It's easy to get caught up in the excitement of Domino and the Notes Web browser—and to forget that the Internet can be used as an easy and worldwide way to connect your Notes servers and workstations. When push comes to shove, the Internet is just a big TCP/IP based Wide Area Network, and using it to connect "Notes to Notes" for purposes of replication, mail routing, and so on can save you a lot of money and work. (It'd be a lot cheaper and easier to connect your organization's worldwide servers via the Internet than it would be to build a global WAN of your own!)

Put a single Domino server outside your firewall, and your Notes workstation users around the world can connect to it directly through the Internet. Enable "Passthru" on that Domino server, and users (as well as other servers) can even securely access the Domino servers that are behind the firewall. For applications that do not require such tight security and authentication, Domino servers on the Internet also let you "publish" Notes databases to the world—so that any Notes workstation, anywhere in the world, can connect to a server and use the databases on it. This presents some unique opportunities to use Notes as a publishing medium.

## THE BASICS: SETTING UP THE CONNECTION

Notes can connect to the Internet in two ways: through an Internet Service Provider (ISP), or through a Notes Public Network (NPN). ISP connections can range from dial-up SLIP connections (cheap) to full-time leased line connections (expensive).

Notes Public Networks are essentially huge WANs that have been optimized for Notes and that are offered by (for a fee, needless to say) telecom carriers such as AT&T, CompuServe, and IBM, among others. Using a NPN lets you outsource the hardware and support required to run a large, worldwide WAN. (A company called WorldCom deserves special notice, as they specialize in Notes services, and the author's personal experiences with this company have always been excellent.)

### Setting Up a LAN to Connect to the Internet through an ISP

If a Domino server is intended to deliver mail and replicate databases with another server through the Internet, a leased-line connection from a router on your local area network to an Internet Service Provider's wide area network provides the round-the-clock connectivity required in such high-traffic environments. Applications requiring less than constant connection can connect to an ISP via a dial-up SLIP connection—albeit more slowly. (Some hardware vendors are starting to manufacture routers that can function as dial-up routers—connecting only when they receive a LAN packet destined for the Internet.) Figure 23.1 illustrates how Notes LANs can be connected to the Internet via an Internet Service Provider.

If you have the financial (and, perhaps, intellectual) wherewithal to connect to an ISP with a leased line, install TCP/IP on the router (and the Domino server, of course). A less expensive (and usually easier to install and maintain) course of action takes advantage of dial-up connections, and relies on SLIP/PPP for connectivity.



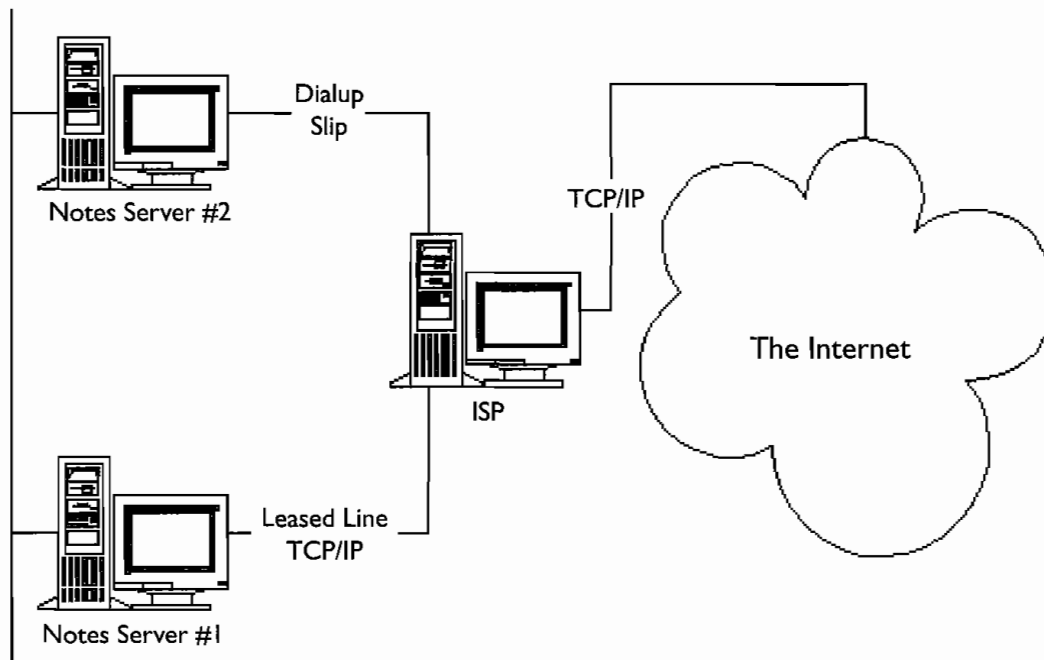


**Figure 23.1** Notes can connect to the Internet via an ISP—through either a leased line or a dial-up connection.

#### Enabling TCP/IP or SLIP/PPP on a Server

Installing the TCP/IP stack of your choice is beyond the scope of this book; the steps involved vary depending on the stack's vendor. For the sake of this section, assume that TCP/IP or SLIP/PPP has been successfully installed on the machine. Specific recommendations about which TCP/IP stack to use, and which vendors are supported by Lotus, can be found in the **Lotus Notes Network Driver** documentation database.

There are two steps to configuring a Notes server to use TCP/IP. First, enable the TCP/IP on the server, and then edit the server's Server document in the Public N&A Book to include the new protocol.

To enable the TCP/IP port, choose **File+Tools+User Preferences** and click the **Ports** button. Select **TCPIP** in the Communication Ports list, and then click the **Port Enabled** check box. You can use the **TCPIP Options** button to increase the TCP/IP timeout from the default value of five seconds. Figure 23.2 shows the User Preferences Ports dialog box.

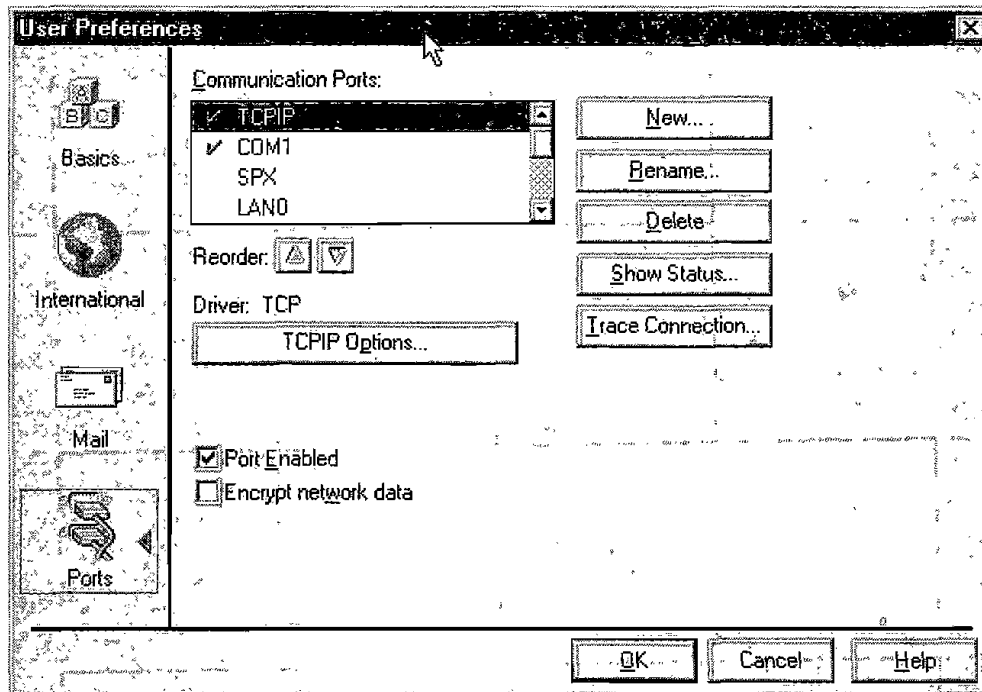


**Figure 23.2** Enabling the TCP/IP port on a Notes server (or workstation).

To allow other servers and workstations to connect to the server through the newly-enabled TCP/IP port, edit the server's Server document in the Public N&A Book. Add the name of the port you have enabled in the Port field, the name of the Notes Network to which the server belongs in the Notes Network field, and the server's numeric TCP/IP address in the Net Address field.

**WARNING**

Entering the wrong TCP/IP address into the Net Address field can have disastrous results. If you're not sure what to enter into this field, contact your organization's TCP/IP administrator, or your ISP's customer service department.

### Enabling TCP/IP or SLIP/PPP on a Workstation

There is only one step to turn on TCP/IP for a Notes workstation; once you've enabled the port, the workstation is ready to roll. To enable the TCP/IP port on a Notes workstation, choose **File+Tools+User Preferences** and click the **Ports** button. Select **TCPIP** in the Communication Ports list, and then click the **Port Enabled** check box. As with a Notes server, you can use the **TCPIP Options** button to increase the TCP/IP timeout from the default value of five seconds.

### Using Notes Public Networks to Connect to the Internet

Depending on the size of your organization, and the scale of your requirements for a Notes infrastructure, there may be several advantages to using an NPN to provide your WAN connectivity. If you rely on an NPN, the majority of your servers are physically located at the telecom provider's location, so you (obviously) save space. More importantly, consider the reduced administrative burden and costs associated with outsourcing these services: at the very least, you don't have to hire, train, and house people to run the Notes WAN. NPN carriers typically publish service level agreements guaranteeing minimum down-times, minimum throughput, etc., so using an NPN can also increase the overall performance and reliability of your Notes WAN.

On the other hand, it costs money to use an NPN to deliver Notes services, so such a scheme makes sense only for medium-to-large organizations. Smaller shops may be able to get by with a few Notes servers that use the Internet for WAN connectivity; Figure 23.3 illustrates how an NPN could be used to connect Notes servers within your organization.

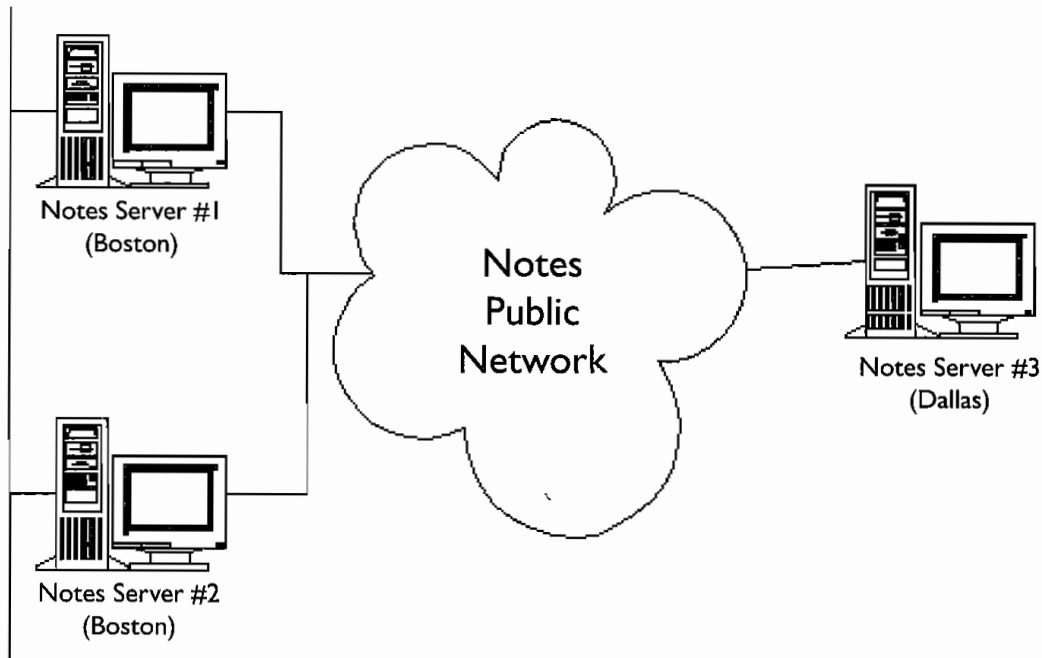


**Figure 23.3** Outsourcing Notes services to a Domino Public Network.

More information about NPN carriers, services, and rates is available via Lotus Development's Home Page, the URL for which is <http://www.lotus.com>.

#### Connecting a Workstation to a Server through the Internet

You can use a Notes workstation to connect to a Notes server that's on the Internet, assuming:

- The workstation and server have both been configured to run TCP/IP (in other words, the TCP/IP port has been enabled)
- You know the Notes server's IP address
- The security settings and certificate(s) on the Notes server are set to allow the workstation access

For example, if a member of the sales team was on the road, he (or she) could dial up to a local ISP, and then use the Internet to connect to the Notes server back at the office. Since it is often cheaper to dial up a local ISP

than it is to place a long-distance phone call to the Notes server directly, you just might save some money. Figure 23.4 shows how this fits together.

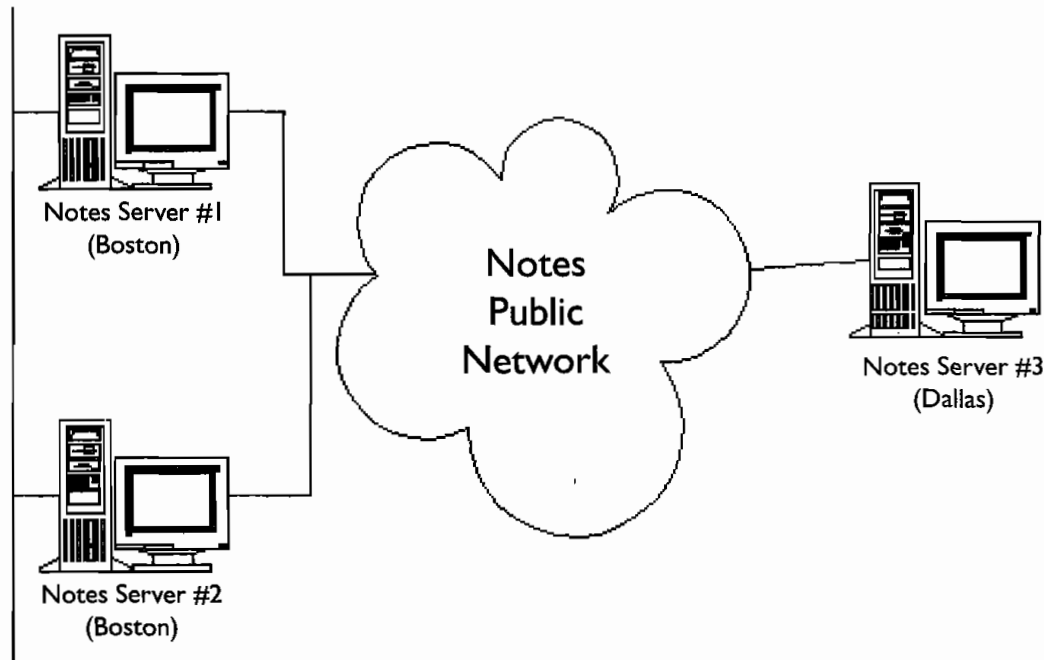


**Figure 23.4** Notes workstations can connect to Domino servers over the Internet.

Also note that if the Notes server that's on the Internet has been configured as a Notes Passthru server, the remote workstations can also connect to the other servers (that are probably behind the firewall). The Lotus Notes Help database contains information about configuring Passthru servers.

#### *Creating a Location Document on a Workstation*

Once the TCP/IP port on the workstation has been enabled, must you add a new Location document (or edit any existing one) to take advantage of the new port? Not true. I added a port in 144.5 and it was automatically enabled in my location document. Same thing with just enabling it. Choose **File+Mobile+Locations** and then use the **Edit Location** or **Add Location** button in the Action bar. To make use of the newly-enabled TCP/IP Port, check off the Port name in the **Ports to use:** section of the form, as shown in Figure 23.5.

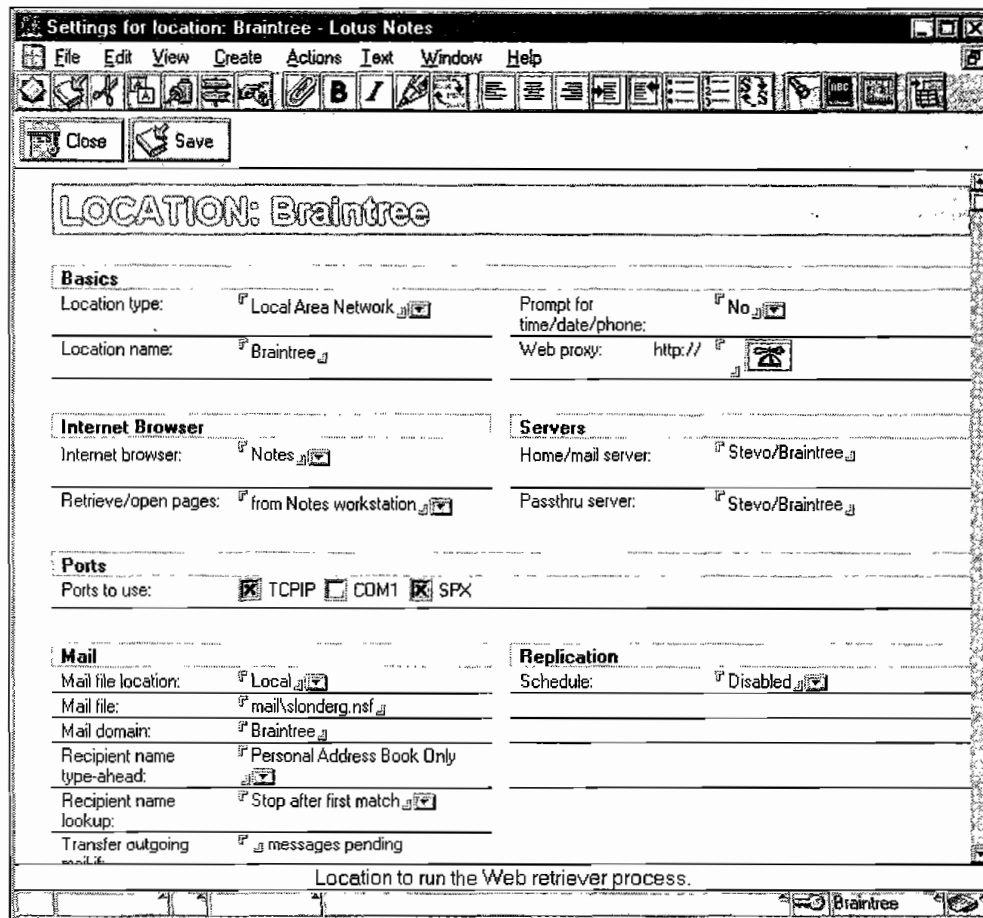


**Figure 23.5** Edit a Location document on a workstation to use the new TCP/IP port.

#### *Creating a Connection Document on a Workstation*

In the absence of a DNS on the LAN (or a local hosts file on the workstation), a Connection document is required that includes the name and TCP/IP address of the Notes server with which the workstation will be connecting.

To add a new Connection document to the Personal N&A Book on a Notes workstation, choose **File+Mobile+Server Phone Numbers**, and then click the **Add Connection** button in the Action bar.

Four fields are required for the workstation's Connection document:

- Connection type, which is **Local Area Network**,
- Server name, in which you enter the name of the Notes server you will connect to through the Internet;
- Use LAN port, in which you choose **TCPIP**.
- Destination server address, in which you enter the (numeric) TCP/IP address of the Notes server you entered into the Server name field.

Figure 23.6 shows a sample Connection document for a workstation that will be connecting through the Internet to a Notes server called **Sales\_1**.

The screenshot shows the 'Server Connections' window in Lotus Notes. The window title is 'Steve's Address Book - Server\Connections - Lotus Notes'. It has a menu bar (File, Edit, View, Create, Actions, Section, Window, Help) and a toolbar with various icons. Below the toolbar are buttons for 'Add Connection', 'Edit Connection', and 'Delete Connection'. A left sidebar shows a tree view with 'Locations', 'People', 'Server' (selected), 'Certificates', and 'Connections'. The main area displays a table with columns 'Server', 'Port', 'Via', and 'For User'. The first row shows 'Sales\_1/Braintree', 'tcpip', '206.119.10.89', and 'Steve Londergan'. Below the table is an 'Edit Connection' button. The main content area is titled 'SERVER CONNECTION: Sales\_1 206.119.10.89'. It contains two sections: 'Basics' and 'Advanced'. The 'Basics' section has 'Connection type: Local Area Network' and 'Use LAN port: tcpip'. The 'Advanced' section has 'Only from Location(s): \*', 'Only for user: Steve Londergan/Braintree', 'Usage priority: Normal', 'Destination server address: 206.119.10.89', and a 'Comments' field. The status bar at the bottom shows 'Braintree'.

| Server            | Port  | Via           | For User        |
|-------------------|-------|---------------|-----------------|
| Sales_1/Braintree | tcpip | 206.119.10.89 | Steve Londergan |

**SERVER CONNECTION: Sales\_1 206.119.10.89**

**Basics**

Connection type: Local Area Network  
Use LAN port: tcpip

**Advanced**

Only from Location(s): \*  
Only for user: Steve Londergan/Braintree  
Usage priority: Normal

Destination server address: 206.119.10.89  
Comments:

**Figure 23.6** A Connection document in a workstations Personal N&A Book tells the workstation how to connect to the Notes server through the Internet.

### Accessing a Notes Server from a Notes Workstation through the Internet

Connecting to a Notes server through the Internet is no different from connecting to any other Notes server, assuming you have completed the previous two steps of adding a Location and a Connection document to the workstation's Personal N&A Book. Choose **File+Database+Open**, type (or choose) the name of the Notes server in the Server field, and you're in business. Of course, depending on the TCP/IP or SLIP/PPP software you use, you may have to establish the network connection before you use the File+Database+Open dialog box.

**NOTE**

As with any Notes workstation-to-server connection, the Access Lists, Restrictions, and Certificates on the server must be defined properly to allow a workstation access.

### Connecting a Server to another Server through the Internet

Just like a workstation, a Notes server can connect to another Notes server through the Internet, assuming:

- Both of the servers have been configured to run TCP/IP (in other words, the TCP/IP port has been enabled)
- You know the Notes server's IP address
- The security settings and certificates on the two Notes servers are set to allow one another access

As we have seen, using the Internet to connect Notes servers is a great way to connect remote servers without having to dial the servers directly, lease lines between the servers, and so forth. Notes servers can use the Internet to perform such routine server tasks as mail routing, SMTP mail delivery, and replication. Figure 23.7 shows how this fits together.



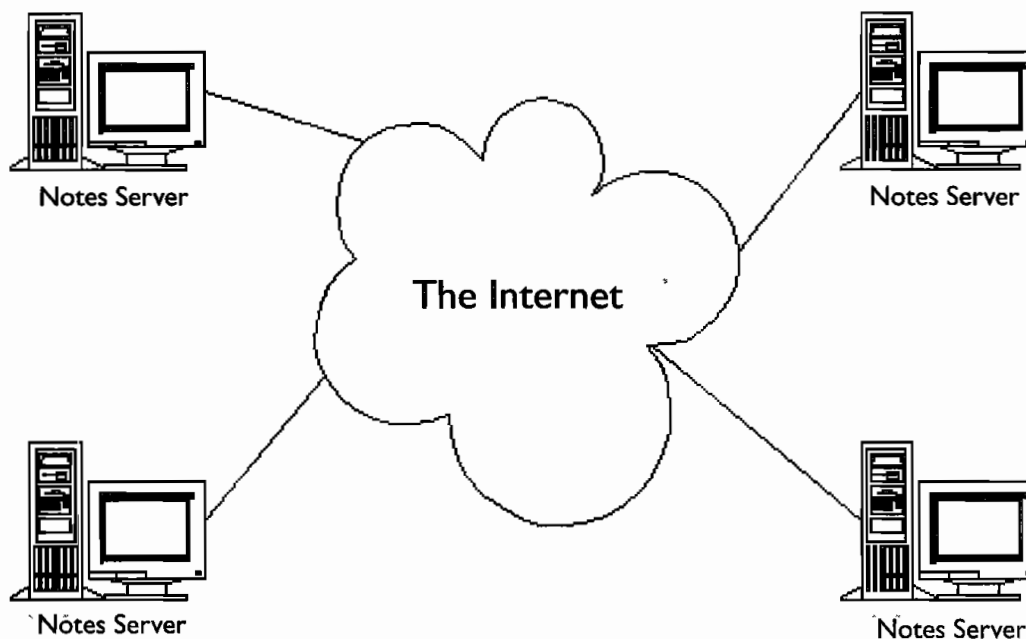


**Figure 23.7** Connecting Domino servers to one another through the Internet

As with a Notes workstation, configuring a Domino server to be able to make connections through the Internet is a relatively simple matter of creating a Connection document in the server's copy of the Public N&A Book.

*Creating a Connection Document for a Notes Server*

To add a new Connection document to the Public N&A Book on a Notes server, start the Notes workstation software *on the server* and choose **File+Mobile+Server Phone Numbers**, and then click the **Add Connection** button in the Action bar. It's worth noting that creating a Connection document for Internet connections is no different from creating a Connection document for regular LAN connections.

Nine fields are required for the server's new Connection document:

- Connection type, which is **Local Area Network**;
- Source server, into which you enter the name of the Notes server that will be initiating the Internet connection;
- Source domain, into which you enter the name of the Notes domain to which the Source server belongs;
- Use the port(s), into which you enter **TCPIP**;
- Destination server, into which you enter the name of the Notes server with which the Source server will be connecting;
- Destination domain, into which you enter the name of the Notes Domain to which the Destination server belongs;
- Optional network address, which is not optional at all, since you must enter the (numeric) TCP/IP address of the Destination server, unless it is in the DNS or local hosts file;
- Tasks, into which you choose the server tasks for the Source server that can take advantage of the Internet connection—usually **Routing and Replication**.

The Tasks field tells the server *why* it is connecting to the Destination server, and the fields in the Scheduled Connection and Routing and Replication sections of the Connection document tell it *when* it will connect to the Destination server. As we have seen, determining the schedule for the connections between the two Internet servers is no different from determining the schedule of connections between two LAN-based Notes servers.

Figure 23.8 shows a sample server's Internet Connection document.

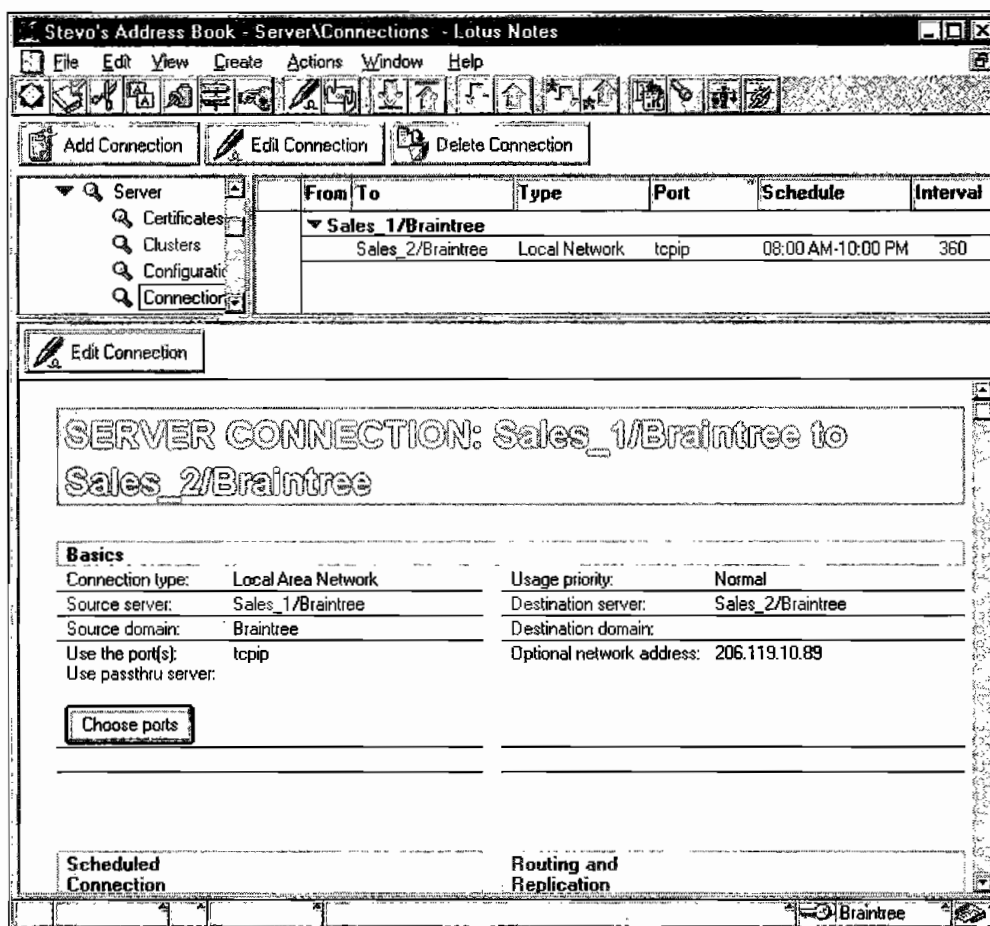


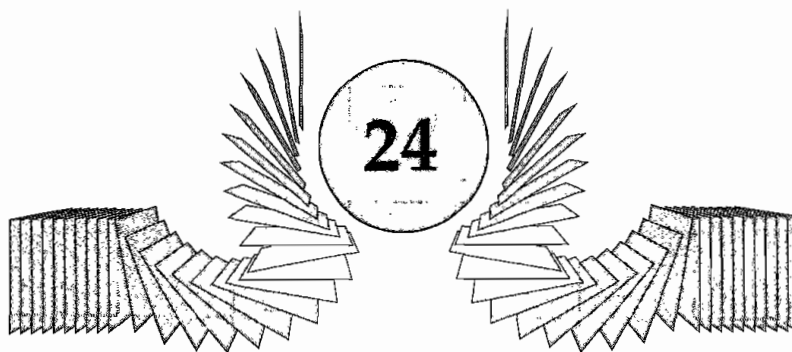
**Figure 23.8** A Connection document in a server's Public N&A Book tells the server how and why to connect to the another Domino server through the Internet.

#### Accessing a Domino Server from a Domino Server through the Internet

The Source server will connect automatically to the Destination server, based on the tasks and schedule you define in the Connection document. Depending on the TCP/IP software that's installed on the Notes server, it may be necessary to manually establish the TCP/IP connection before starting the server. Some TCP/IP stacks will load and connect automatically whenever the Source server begins to initiate a connection.

**SUMMARY**

- Notes workstations and Domino servers can use the Internet to communicate with one another.
- Use the **File+Tools+User Preferences+Ports** dialog box to enable the TCPIP port on a Notes server or workstation.
- The Certificates, Access Lists, and Server Restrictions must be set properly to allow a workstation or server to access a Notes server through the Internet.



# **SECURITY CONSIDERATIONS FOR DOMINO SERVERS ON THE INTERNET**

Security is often the first consideration when configuring a server for the Internet; specifically, finding a balance between security measures that will ensure that a server is safe from unwanted visitors, while at the same time making the system easy to use for legitimate users. Domino servers that are used to retrieve Web pages for users, or that provide access to Notes users via the Internet, are particularly “securable” by combining the built-in security features of Domino servers—such as Server Access Lists and restrictions with LAN-based Proxy servers and Firewalls.

In this chapter, we’ll provide an overview of the security measures that are built into Domino servers, as well as some practical advice and examples about configuring Domino servers to use Firewalls and Proxy servers to ensure system integrity and privacy.

## **DOMINO SECURITY BASICS**

Given the importance of the topic, it is worth a short summary of the security features that are built into Domino servers:

- **Authentication:** Domino servers use an industry-standard, challenge-response transaction to ensure that the user “really is who he says he is.” Passwords and User Names (for Web users) are stored in the Public N&A Book, while the User Name, Password, and Certificate for a Notes workstation user is stored in his USER.ID file. Domino servers also support, when appropriate, Anonymous access and un-authenticated server access.
- **Access Control Lists (ACLs):** Every Notes database maintains an Access Control list that is used to determine who can do what in the database/Web area. Of particular importance to the Domino Webmaster are the Anonymous and Default entries, which can be used to allow or prevent Web users “at large” from accessing a particular database, or a particular View, Form or document within a database.
- **Channel encryption:** When Domino servers connect with one another over the Internet for the sake of Replication or Mail routing, they can take advantage of channel encryption to guarantee that the data is secure while being passed between the servers. (This feature can also be used when Notes workstation clients connect to Domino servers over the Internet.)
- **Roles:** Working in conjunction with and as a refinement of a database’s ACL, a Role can be used to assign specific access to certain parts of a Web site or set of Web pages.

## DOMINO APPLICATION-LEVEL SECURITY

In general, Domino servers are connected to the Internet for one of four reasons:

1. To act as a Public Web server
2. To act as a Private Web server to which employees connect
3. To act as a Public server for Notes clients
4. To act as a Private server for Notes clients

Depending upon the reason the Domino server is connected to the Internet, there are several mechanisms built into Domino servers to control who has access to what.

### A Public Domino Server for Browser Users

When a Domino server is used to publish Web pages for consumption by the general public, the Domino server is, by necessity, connected directly to the Internet.

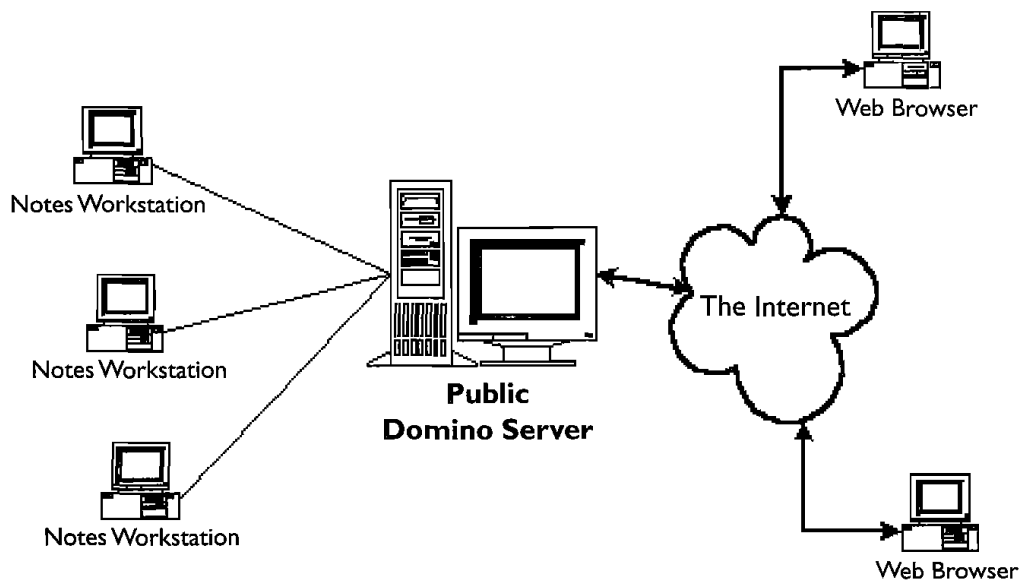


**Figure 24.1** A typical configuration for a Public Domino server.

As we discussed in Chapter 6, database-level access on a Domino server is configured via database Access Control Lists, which in turn rely on User Names and Passwords that are defined in the Domino server's copy of the Public N&A Book. Typically, a public Domino server's database ACLs provide varying degrees of access. For example, the database containing the site's Home Page might allow any Web user access, while other databases on the Domino server would rely on a registered User Name for access.



For a Domino server to be public, the `Allow Anonymous HTTP Connections` field in the server's Server document must be set to **Yes**.

## A Private Domino Server for Browser Users

When a Domino server is used to publish Web pages for consumption *not* by the general public, the Domino server is also connected directly to the Internet.

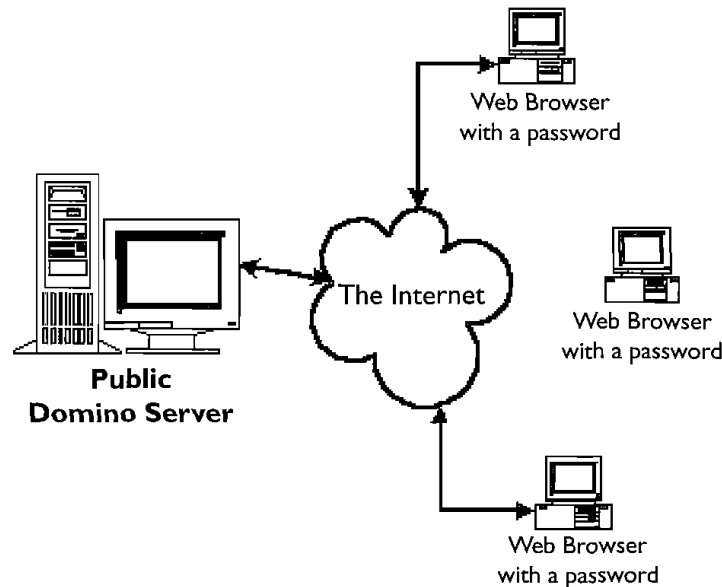


**Figure 24.2** Private Domino servers that are connected to the Internet rely on authenticated user names, database ACLs, and the server's Security settings in the N&A Book to keep the server private.

In this scenario, registered User Names and Access Lists become much more important, as they are the mechanism by which the server and its contents are kept Private. The most direct way of keeping a Domino server Private is to set the `Allow Anonymous HTTP Connections` field in the server's Server document to **No**, in which case the Domino server will force all users to enter a valid, registered, User Name and Password



when contacting the server. In addition to requiring users to authenticate upon contacting the Domino server, each of the databases ACLs should also be configured so that the `Default` setting allows for **No Access**.

### A Public Domino Server for Notes Clients

A Domino server is also connected directly to the Internet when the databases on it are intended to be publicly accessible by Notes workstation users.

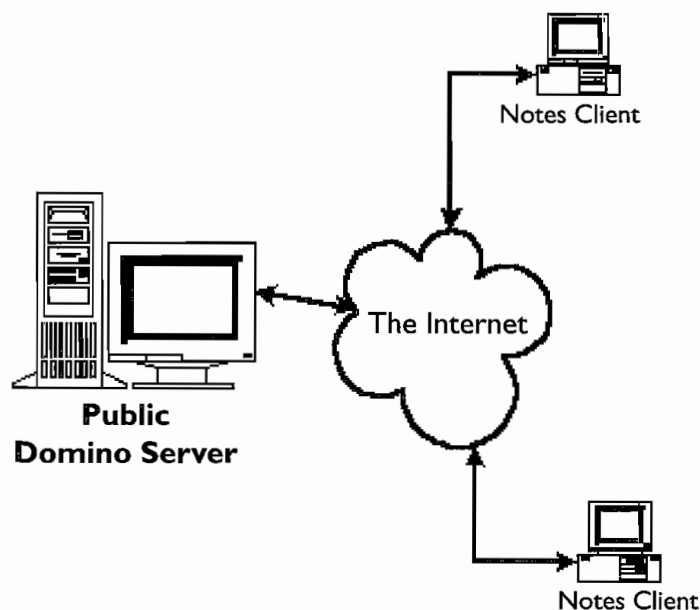


**Figure 24.3** A Public Domino server providing access to Notes clients.

In this scenario, users with the Notes client software can connect to the Domino server via the Internet to access the databases on it. For an unauthenticated Notes workstation user to be able to use a Domino server via the Internet, the `Allow Anonymous Notes Connections` field in the server's `Server` document must be set to **Yes**, so that the Domino server will not require users to have a valid `Certificate`. Additionally, the databases' ACLs on the Public server must also contain an entry for `Anonymous`. (In most cases, `Anonymous` and unauthenticated Notes users are not allowed access higher than `Reader` to such databases.)

For a Notes client to be authenticated by a Public Domino server being accessed via the Internet, the workstation's USER.ID file must contain a Certificate in common with the server's.

### A Private Domino Server for Notes clients

When Notes workstation users use the Internet to connect to a Domino server, the server is connected directly to the Internet.

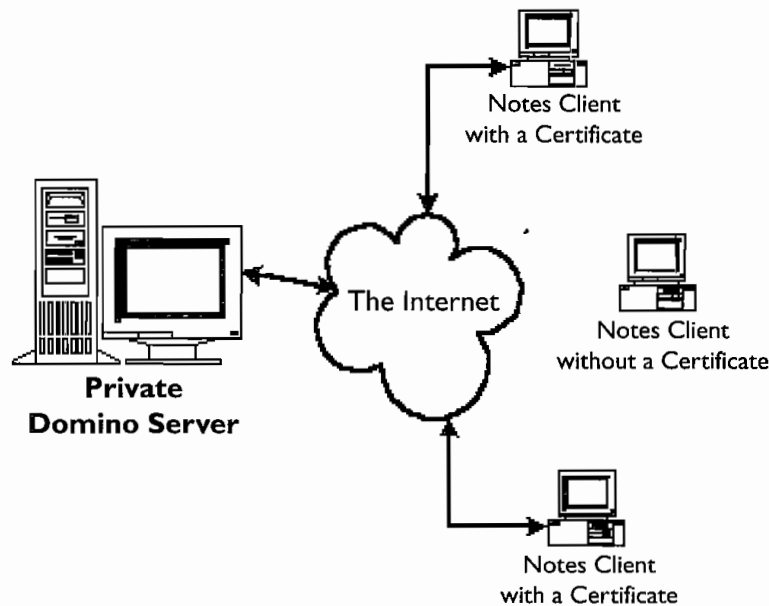


**Figure 24.4** Private Domino servers providing access for Notes workstation users rely on common certificates, authenticated user names, database ACLs, and the server's Restrictions settings to keep the server private.

The "first line of defense" for a Public Domino/Notes server is the Certificate that is required to be shared between all users of the server; setting the Allow Anonymous Notes Connections field in the server's Server document to **No** forces the Domino server to automatically authenticate all users of the server, thereby immediately "weeding out" and denying access to any user who does not have a valid Certificate for the server.

Additionally, each of the databases ACLs would also be configured so that the Default setting allows for **No Access**.

## DOMINO NETWORK-LEVEL SECURITY

There are two ways that Domino servers that are connected to the Internet can be configured for network-level security:

1. With an existing Firewall server.
2. Configuring the Domino server itself as an application-level Firewall server.

### Domino, Firewalls, Proxies, and You

A *Firewall server* is a separate computer that sits between a private LAN and the Internet, and acts as a kind of filter between the two networks. In many cases, the Firewall server is configured to filter the network packets that it receives so that only the appropriate information can be passed between the two networks, and would be the only computer that was physically connected to both of the networks. (In other words, it's the only computer in the organization that is actually connected to the Internet.)

For example, a Firewall server might be configured to allow HTTP requests from Web users who are connected to the Internet to pass to a Domino server on a company's internal network, while at the same time denying FTP requests from the Internet—thereby isolating and securing the organization's network nodes from unwanted intrusion.

A Firewall server is also often configured to run as a *Proxy*, which is an application that runs on the Firewall server and presents a "public face" to the Internet; when a user behind the Firewall/Proxy server requests a Web page, the request is passed to the Proxy, which in turn passes the request to the Web server on the Internet. The Proxy receives the Web page back from the Web server, and forwards the page to the user who requested it in the first place. Using a Proxy/Firewall server to secure and separate two networks (such as a corporate LAN and the Internet) gives the administrator a single point of exit and entry for all Internet traffic; managing and securing a single computer that is connected to the Internet is certainly easier than managing and securing Internet connections for every Web user and Domino server on the corporate LAN.

Proxy/Firewall servers also give the network manager the opportunity to manage—and possibly constrain—exactly what kind of LAN traffic can be passed between the two networks. For example, a LAN manager could configure the Firewall/Proxy server such that it would not pass FTP requests to the Internet, or so that it refused all HTTP requests from outside the Firewall.

Finally, Proxy/Firewall servers protect the identity of users and, perhaps, information about the layout of an internal LAN from prying eyes. As we have seen, when a user behind the Firewall requests, say, a Web page, the request is handled by and processed through the Firewall/Proxy server—thereby guaranteeing the anonymity of the user who requested the page in the first place. And since prying eyes would not be able to deduce the IP address of the user, it is impossible to draw any conclusions about a LAN's layout from outside the Firewall.

In the simplest cases, a Firewall server is nothing more than a LAN router configured to isolate two networks from one another. More sophisticated Firewalls (like those that run Proxy software) are actual computers.

#### A Packet-filtering Firewall

Most Firewall systems, including network routers, provide the ability to filter network packets. In this scenario, the network administrator configures the device acting as the Firewall to decide, based on the TCP/IP port over which the packet is being transmitted, whether or not the packet will be allowed to pass through the Firewall.

For example, a LAN router might be configured to allow requests from Internal LAN workstations over port 80—which is the Port for HTTP requests—to pass through the Firewall to the Internet. This allows the organization's users to access Web servers on the Internet. At the same time, the router could be configured to disallow packets on Port 1352—which is the TCP/IP port used for "regular" Notes client/server traffic—and to refuse incoming HTTP requests on Port 80. In this manner, Notes workstation and Web browser users outside the Firewall will not be able to access the internal Domino server.

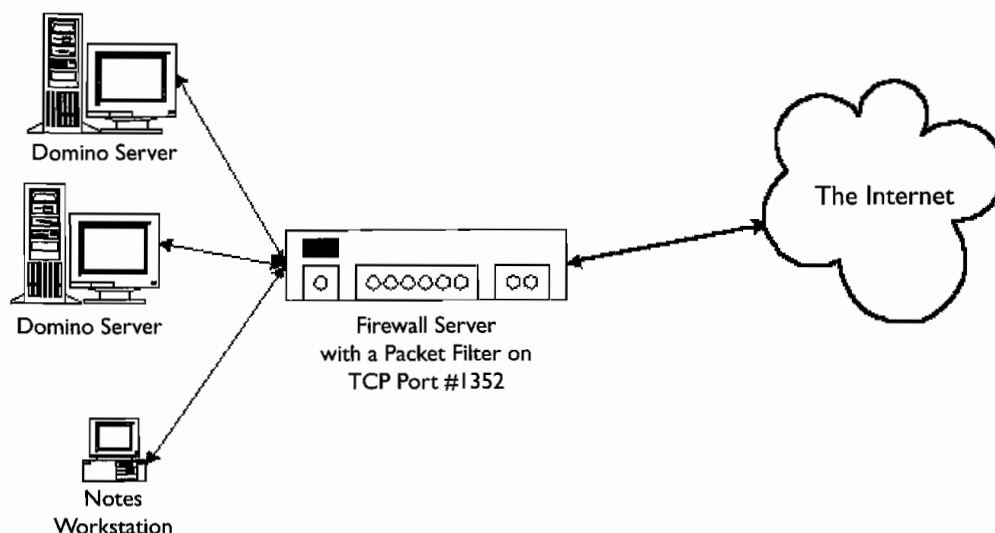


**Figure 24.5** Using a Firewall to filter TCP/IP packets.

### An HTTP Proxy

One special kind of Proxy is known as an *HTTP Proxy*. Like the others, this Proxy allows users on the corporate LAN to access the Internet through the Firewall, but this type of Proxy uses HTTP in all communication between the user and the Firewall server, even if the Proxy server itself is connecting to an FTP or GOPHER server on the Internet. In other words, if a user on the local LAN requested a page on a Gopher server on the Internet, the Proxy will retrieve the page for the user via Gopher, but will use HTTP to send the page to the user.

This type of Proxy offers the distinct advantage of limiting the Internet protocols required within the local LAN, and greatly simplifies the configuration of the Proxy and Firewall. Since all communication between the local users and the Proxy are carried over HTTP, the Firewall server can be configured to exclude all of the other Internet protocols—FTP, WAIS, and so on.

Most HTTP Proxies also support *Secure Sockets Layer* (SSL) for encrypting and authenticating Web traffic. When SSL is being used between a Web user and server via an HTTP Proxy, the HTTP Proxy

doesn't read or otherwise interpret the data being exchanged between the client and the server.

**NOTE**

SSL uses the CONNECT method (as opposed to GET) to open a secure tunnel through an HTTP Proxy.

The architectural layer that is used for all communication between Notes workstations and servers is known as *Notes Remote Procedure Call* (RPC). If an HTTP Proxy will be used to pass Notes-to-Notes traffic between a local LAN and the Internet, it must support the CONNECT method, just like SSL. Configure the Proxy to allow a CONNECT method for port 1352 to allow Notes RPC to pass through the Proxy.

#### A Circuit-level Proxy (such as SOCKS)

Circuit-level Proxies, such as SOCKS, work "above" the TCP/IP Port or application layer of the protocol, and allow clients to pass through a central service on the local LAN to connect to whichever TCP/IP Port the user requests. Since SOCKS servers have the ability to authenticate the source address of a connection request, a SOCKS server can be used to prevent unauthorized nodes from accessing the Internet. For example, a SOCKS Proxy server could be used to allow local users to connect to Web servers on the Internet, but prevent the local Domino servers from issuing HTTP requests destined for Web servers outside of the local LAN.

Notes workstations and Domino servers automatically have built-in support for SOCKS Version 4.2, and will usually interoperate with other servers running SOCKS Version 5. Notes workstations and servers that will be accessing the Internet (or communicating with one another) via a SOCKS server must employ a TCP stack that supports transparent SOCKS transactions.

### Using a Domino Server as an Application-level

Even when you do not have an external server, you can still secure Notes and Domino transactions by employing the Domino server itself

as a solution to provide secure access to the Internet for Notes clients and Domino servers.

#### Using Multiple LAN Cards on the Domino Server

An extremely simple and cost-effective way to employ a Domino server as a is to use two LAN cards/adapters in the machine; one card runs TCP/IP, and is connected to the Internet, while the other card/adapter uses a protocol local to the corporate LAN (such as SPX or NetBIOS).

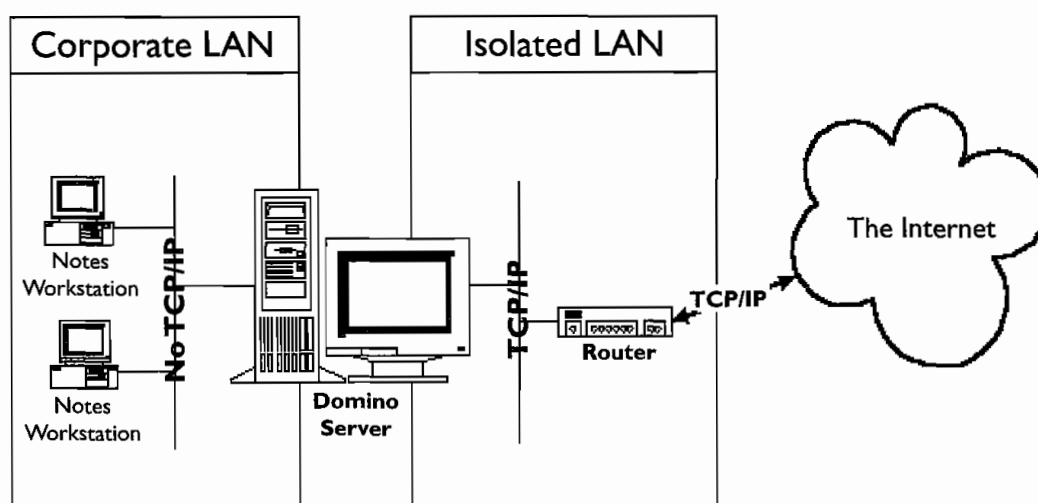


**Figure 24.6** Using a Domino server with two Network Cards as a Server.



If you need to run TCP/IP on the local LAN and on both of the Domino server's network adapters, you must disable TCP/IP routing between the two adapters on the Domino server; and also remove any TCP utilities (like FTP, Sendmail, NFS, etc.) from the server.

#### Using Modems between Domino Servers

An even cheaper—but admittedly slower—method of employing a Domino server as a calls for a modem X.PC connection (or a null modem) between the Domino server(s) and Notes workstations on the local LAN and the Domino server that is connected to the Internet.

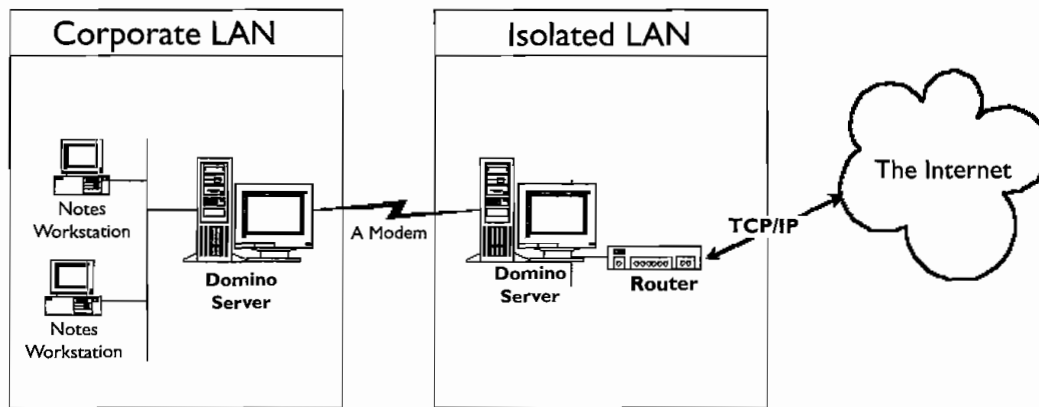


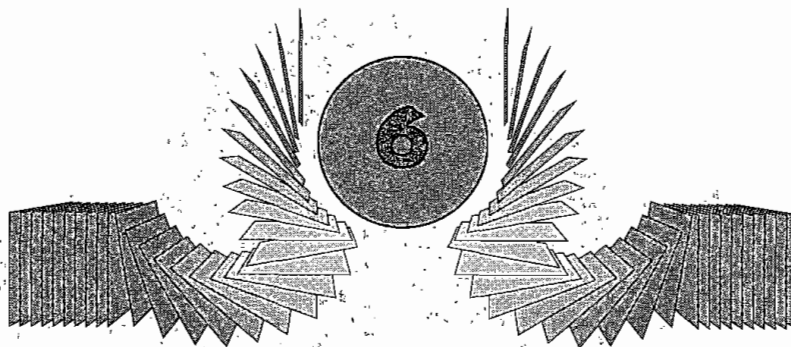
**Figure 24.7** Using a modem to isolate and secure a Domino server.

## SUMMARY

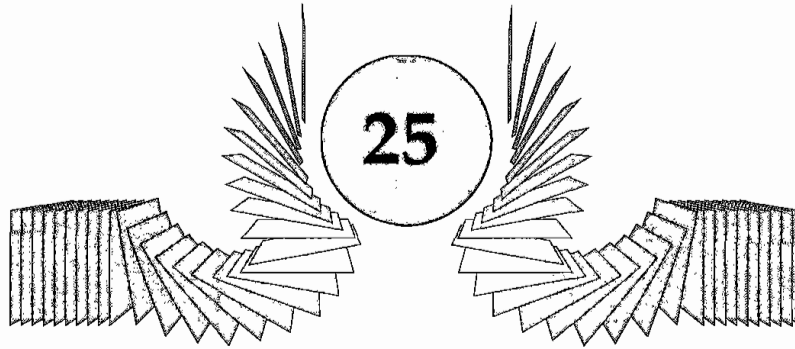
- A Domino server's application-level security is based on database Access Control Lists, Authenticated User Names and Passwords, and the Restrictions settings in the server's Public N&A Book.
- Domino servers can be configured to take advantage of Proxy servers via the Proxy Configuration section of the Server document.
- Notes workstations are also configured to take advantage of Proxy servers via the Proxy Configuration section of the Location document in the workstation's Personal N&A Book.



# SECTION



# DOMINO.ACTION



# **DOMINO.ACTION: THE BIG PICTURE**

As easy as Domino Web site development has been designed to be, Lotus has made it even easier with the advent of Domino.Action. Intended to help you design and assemble the components of a professional-looking Web site "in an afternoon," Domino.Action takes the guess work (and grunt work) out of designing and building an entire Web site.

In this chapter, you'll learn about how Domino.Action Web sites are designed, plus get an overview of the process used by Domino.Action to create a new site.

## **WHAT IT IS**

Domino.Action is a suite of software applications and templates that automates and accelerates the design, development, and management of a professional-looking and solidly-running (Internet or Intranet) Web site. By leveraging the speed with which Notes/Web applications can be built, the extent to which they can be customized with Domino, and the ease-of-use that springs from Lotus' Hi-Test application environment, Lotus has provided a way to quickly and easily build a Web site. It's worth mentioning that the motto for Domino.Action is "build your web site in an afternoon."

Using Domino.Action is a matter of filling out a few forms, answering a few questions, and then letting the built-in agents and programs do the rest. Domino.Action also includes tools so that when you decide that your site needs a cosmetic make-over, or needs a whole new area for, perhaps, a customer survey, the requisite changes to the site are easy to implement. Fill out a few more forms, and Domino.Action creates the necessary Forms, Views, databases, Access Control Lists, entire site areas, and so on.

Domino even lets you choose a “look” for the site, and provides three professionally designed and coordinated looks: *Basic*, *Corporate*, and *Contemporary*. Figure 25.1 shows a sample Home page from a Domino.Action site, using each of the three looks.

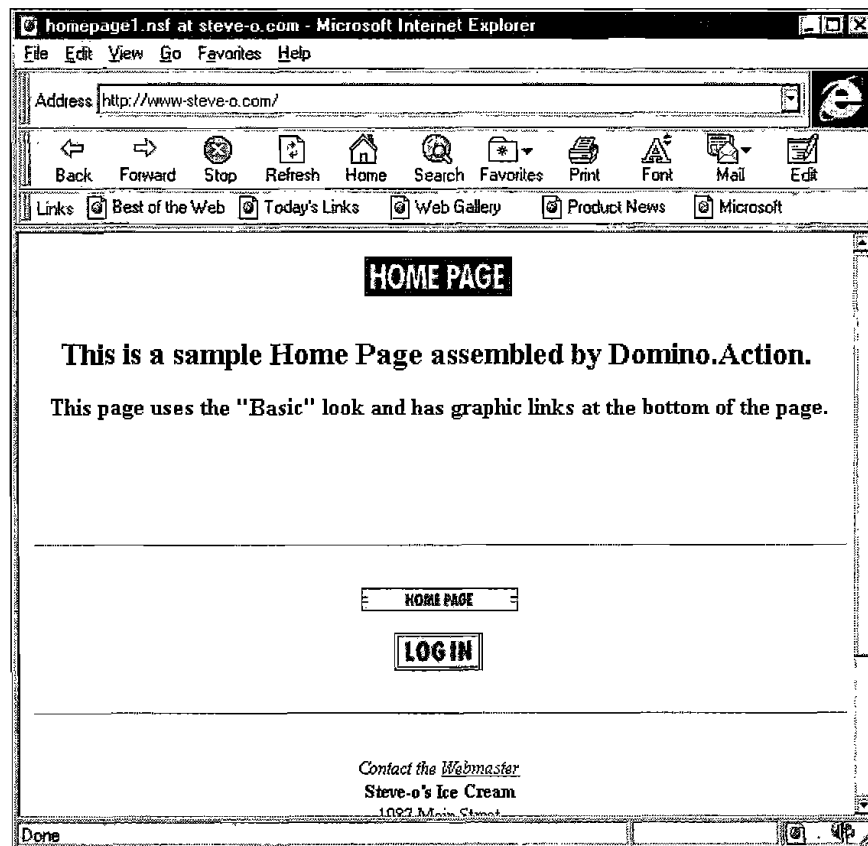


**Figure 25.1** Domino.Action sites can use one of three looks: This is the one called “Basic.”

Domino.Action builds in the roles and responsibilities for content development, so that the site can be configured to allow anyone to add content, or perhaps to allow just the Notes users to add content, or just the Web users, or some mix of both. The built-in workflow processes make auditing and approving documents as simple or as complex as an application's requirements dictate. Domino.Action sites include distributed authoring, editing, and approval workflow processes; robust security; integrated full-text search engine; and document management and aging, so the Webmaster can control who adds which pages and can tell when they were last updated.

## WHAT IT DOES

Domino.Action is made up of three parts: the SiteCreator, the Library, and the AppAssembler.

SiteCreator is the primary application in Domino.Action, and the part that you interact with the most. It is really "just" a Notes database that takes you through the step-by-step creation process, and has the forms that you fill out to specify the naming conventions, page layout styles, custom graphics, and areas for your site.

The Library is also a Notes database, but you do not interact with it directly. This is where Domino.Action stores the "under the hood" elements that it uses as the basis of your site. It is used to store the fields, Forms, database Views, and so forth that Domino.Action uses.

AppAssembler is an external Hi-Test application that performs the consolidation; it uses the Forms you have filled out in the SiteCreator database to determine which of the database elements in the Library database it will use to create the areas of the Web site. It is, in the simplest terms possible, the software program that actually does the work of building the Web site. Figure 25.2 illustrates how the parts of Domino.Action fit together.

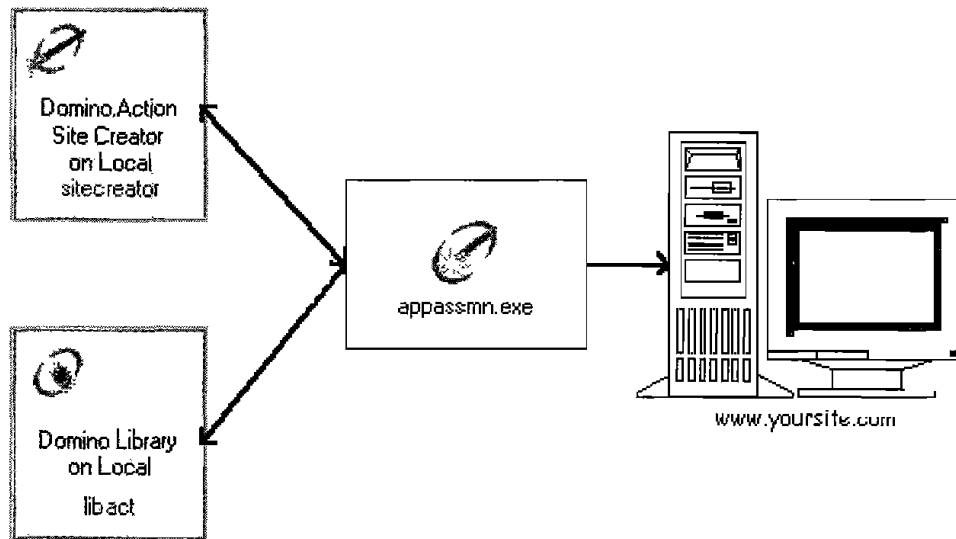


**Figure 25.2** Domino.Action has three parts: the SiteCreator database, the Library database, and the AppAssembler software.

## How It Does It

There are three steps to building a Web site with Domino.Action: (1) configuring the site, (2) designing the site, and (3) generating the site. Each of these three steps is initiated via the Forms, Views, and Agents in the SiteCreator database.



**Figure 25.3** The SiteCreator database.

## Step #1: Configuring the Site

Perhaps the most important part of the site configuration is telling the SiteCreator which areas the site will include. Domino.Action can base on a Web site on any combination of eleven areas:

- **About the Company Area**, which can be used to include documents that describe the company, such as when it was founded, corporate history, etc.;
- **Corporate Policies and Procedures**, especially useful on Intranet-based sites, this area would typically include documents describing a company's standards and practices;
- **Discussion Area**, an interactive and free-wheeling way for a company to interact electronically and directly with its customers;
- **Document Library**, which might contain other information about a company, its products, or services;
- **Feedback**, which allows you to define survey-type questions to be asked of current or prospective customers and employees;
- **Frequently Asked Questions**, or FAQs;
- **Home Page**, the required entrance into the other areas of the site, this part includes a link to all of the other areas you choose;
- **Job/Career Opportunities**, to recruit employees via the site;
- **Product/Services Area**, to create a catalog of Web pages that describe the products and services that the company or department provides;
- **Registration Area**, to automate the registration for users of the site;
- **White Paper Area**, which could hold position papers and the like.

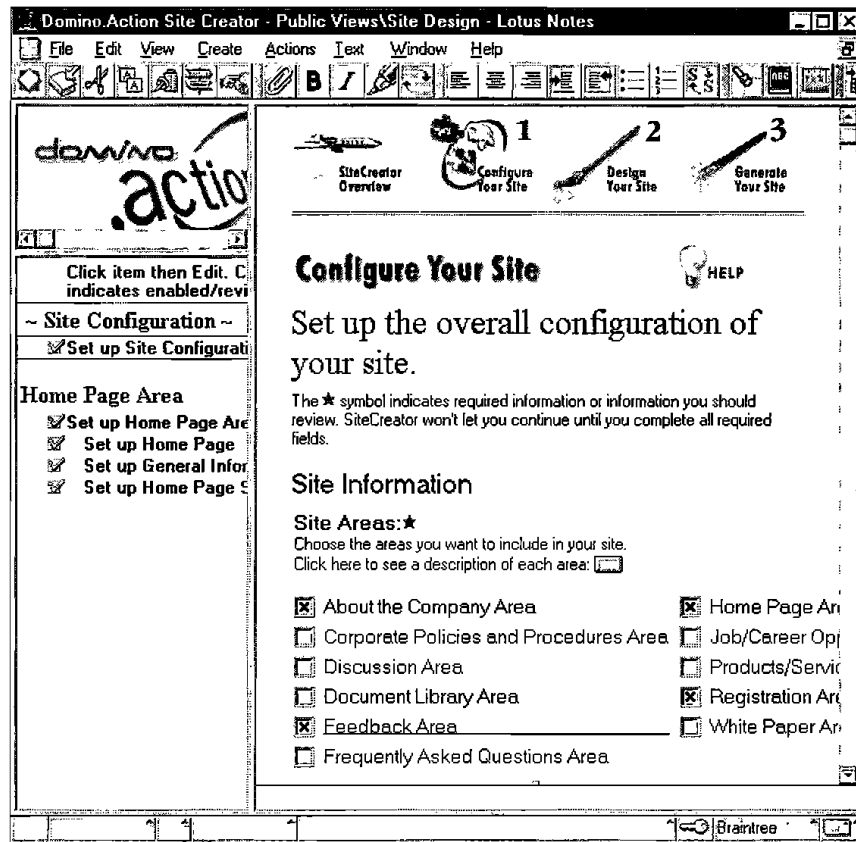


**Figure 25.4** Configuring the site.

During Step #1, you are also prompted to define such essentials as the name of the site, the name of the site's Webmaster, any customized graphics that will be included in the areas of the site, and other such "global" settings.

## Step #2: Designing the Site

To design the site, you answer a series of questions about each of the areas you chose to create in Step #1. For example, if you choose to include a FAQ area, you are prompted to enter information peculiar to that area of the site, such as the "look and feel" of the area's main page, the background color for each of the content pages, as well as the approval process (if any) that new documents go through before being made available on the Web. Refer to Figure 25.5.

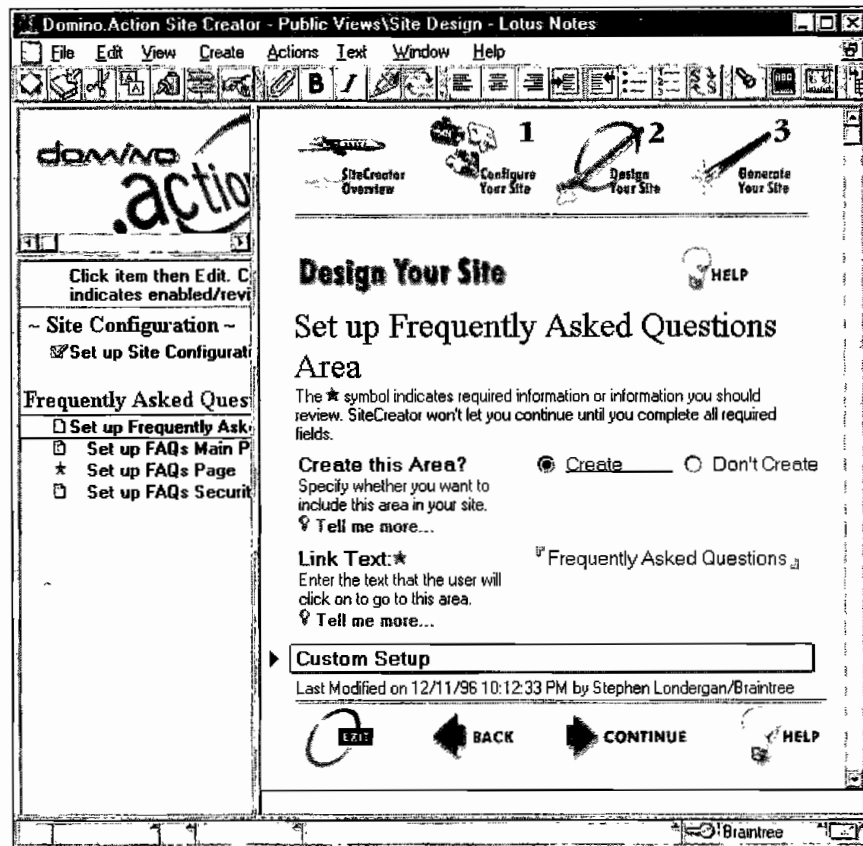


Figure 25.5 Designing the FAQ area.

The actual questions you are asked for each area depend upon its purpose; defining a Feedback area, for example, produces a series of questions in which you define the questions that will be asked of the site's users, while indicating that the site should include a Registration area produces questions like the name of the N&A Book group to which registered members will belong, and so forth.

In general, though, each area has three parts that need to be designed:

- The **Area** profile, which defines the file name for the area's database, and the Link text that visitors click to go to that area.
- The **Main Page** profile determines the look of the page that opens when a Web user opens the area. The main page in any one area automatically includes Links to other pages in the area, as well as Links to all of the other areas in the site.



- The **Security/Approval** profile is used to specify which users can add documents to the database, which user(s) must approve new pages before they are made available to Web users, and who can read the pages in the specific area of the site.

Domino.Action comes with a built-in button that lets you very quickly build the entire site using the default settings for each of the area's look and feel, security settings, and approval requirements. Using this **Quick Design** can have your site up and running and less than an hour—and provides you with the ability to customize and modify the areas that need more than the default settings.

### Step #3: Generating the Site

The third, and final, step to creating a Domino.Action-based Web site is to generate the site. Although you still work in the SiteCreator database, the last button you click "kicks off" the AppAssembler, which is an external program (based on Lotus' Hi-Test programming environment) that does all the grunt work of creating a database for each of the areas you have chosen, building the specialized Forms and Views for each, and then linking the whole pile together to form an integrated and interdependent site.

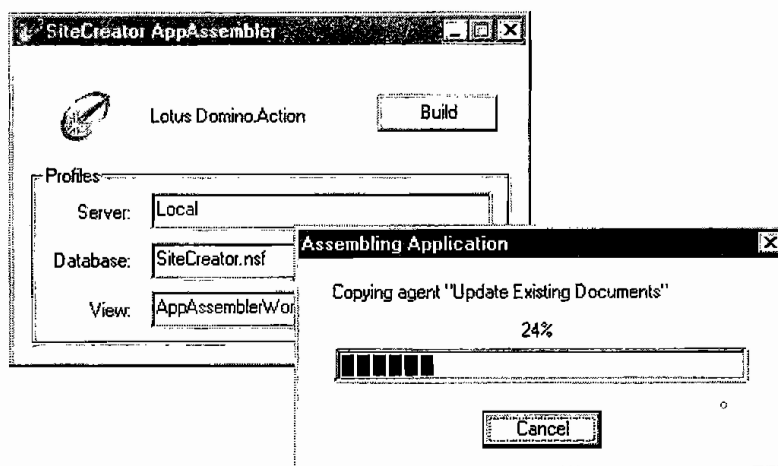


**Figure 25.6** Generating the site.

Depending on how many areas you choose to include, generating the site will take anywhere between from minutes to two hours.

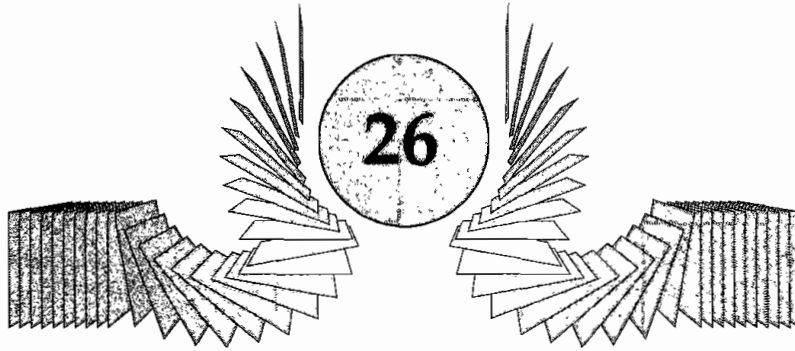
## All the Rest

The SiteCreator database has a few other Views that can be used by the Domino Webmaster to manage Domino.Action. Specifically:

- **Examine SiteCreator Logs:** Each time a site is created, a log document is created to document what the SiteCreator did. The Webmaster can use these logs in the case when a site isn't working, to get a "blow by blow" account of what the program did when creating the site, and presumably detailing any problems that the Hi-Test program encountered while working.
- **Install SiteCreator Executables:** Although the SiteCreator executable program files are installed when Domino.Action is installed, this View of the database can be used to re-install (or perhaps upgrade) the program files.
- **Update Security:** SiteCreator will update the ACL in each area of the site's databases to reflect the approval and page security you have defined in each of the area's Security/Approval documents. This step happens automatically when a site is generated, but you should re-run this step when you have modified any of the site's Security/Approval documents to ensure that the area's ACL matches the new parameters.
- **View Site Images:** This option lets you see all of the images that are included in the site—both the images that come "in the box" with Domino.Action, and also any custom graphics you chose to include while completing the Site Configuration profile.

## SUMMARY

- Domino.Action streamlines and automates the process of building a professional-looking Web site.
- Domino Web sites come in three varieties: Contemporary, Corporate, and Basic.
- There are three steps to building a Web site with Domino.Action: Configuring the site, designing the site, and Building the site.



# **CREATING A WEB SITE WITH DOMINO.ACTION**

The planning of your site's design may be more important than the execution of it. It's especially important to realize that not all Domino.Action sites use all of the available areas; in fact, a few are so alike that they are almost mutually exclusive (the White Paper and Document Library, for example).

In this chapter, you'll learn about planning a Domino.Action site, installing and configuring the SiteCreator and Library databases, as well as the steps involved in generating each type of area, complete with a Security and Approval Profile.

## **PLANNING A DOMINO.ACTION WEB SITE**

Like every other part of a Web site, the key to a successful Domino.Action-based Web site is based on careful planning—but the good news is that the planning required for a Domino.Action Web site is easy. There are just a few key issues to consider before starting to use the SiteCreator database.

### Choose a Look

Each of the areas that will make up the Domino.Action site has associated with it a certain "look," and deciding which a site will use is one of the first questions the nascent Domino.Action Webmaster has to be prepared to answer. The possible choices are:

- Basic,
- Contemporary, and
- Corporate

The actual functionality offered by any one area is not affected at all by the look you choose; rather, Domino.Action uses the look to determine such aesthetic values as the types of icons that are used for Links, their color, style, and so on. In fact, a site made up of, say, three different areas might use a different look for each.

In addition to the look that is used for the pages in each of the areas, Domino.Action also needs to know the Link style that will be used on the pages. For example, the Products and Services area Home Page has Links to all of the other areas on the site, and part of the process of configuring each area is indicating the Link style that the area should use for them. When asked to determine a given page's look, the SiteCreator also prompts for the Link type. These Links can be located at either the top, bottom, or side of each page in the area, and can be based on text, bullets, or graphics.

**WARNING**

As always, the standard advice concerning balancing pretty pictures vs. application performance applies. In particular, Webmasters creating a site with more than two or three areas should use either the **Bullet** or **Text** style Links to increase the speed with which the pages can be retrieved from the Domino server (by reducing the number of images on each page).

The last part of defining each area includes determining which, if any, image files will be used for Form backgrounds.

### Decide on Which Areas the Site Will Include

Domino.Action comes ready to prepare a Web site with up to eleven areas, each of which resides in its own database on the Domino server. Domino.Action automatically integrates the areas that are created with its Site Builder: if you specify that the site is to include, say four areas, each of the pages in each of the areas will have Links to the other three areas of the site. If you chose a site based on five areas, each of the areas would have four Links.

Not all Domino.Action sites necessarily include each of the areas; in fact most *don't*. Several of the areas are designed for use in an Intranet-based Web site, and as such have little use to the Webmaster designing a site for "public consumption." At the same time, some of the areas apply only to public sites, and are often omitted entirely when building a Domino.Action site for use "behind the wall." Domino.Action is flexible, so you can decide to add a new area to the site later, and the Site Builder will rearrange the layout of the other existing areas of the site to accommodate the new area.

There are eleven possible areas on a Domino.Action Web site:

1. **Home Page**, which is probably the only area that is required. This area usually ends up being an entire Domino site's Home Page, and consequently its main job is to provide an index of and Links to the other areas of the site. It also contains documents and pages of its own. Figure 26.1 shows the Home Page of a sample Domino.Action Home Page database.

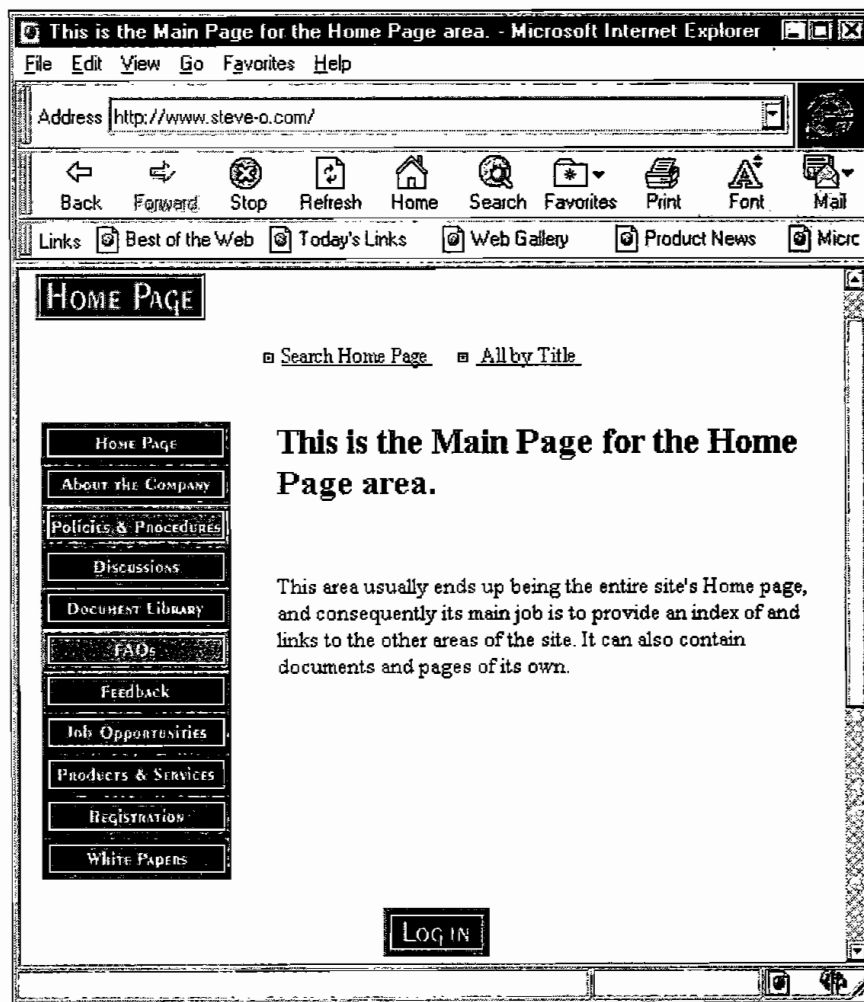


**Figure 26.1** The Main page of the Home Page area on a sample Domino.Action Web site.

2. **About the Company** is used to create and maintain documents about the company, such the corporate mission statement,

company profiles, news articles, press releases, articles about the company's officers, and "Our Company in the News" documents. This area is most often included in sites that are available to the general public on the WWW.

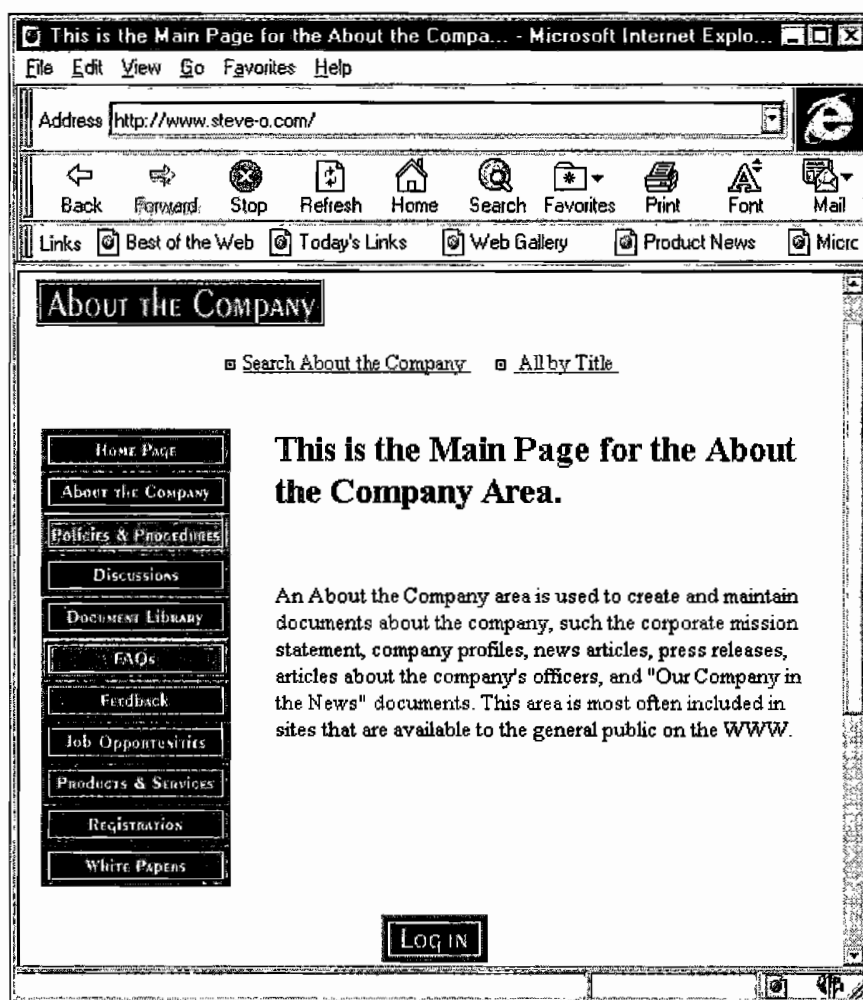


**Figure 26.2** The Main page of the About the Company area on a sample Domino.Action Web site.

3. **Policies and Procedures** holds documents outlining a company's Procedures and Policies, such as sick days, holiday schedules, and so on. Given the internal nature of the documents that are usually stored in this area, it is most often included in Intranet-based Web sites. This area automatically categorizes documents by their type and scope.

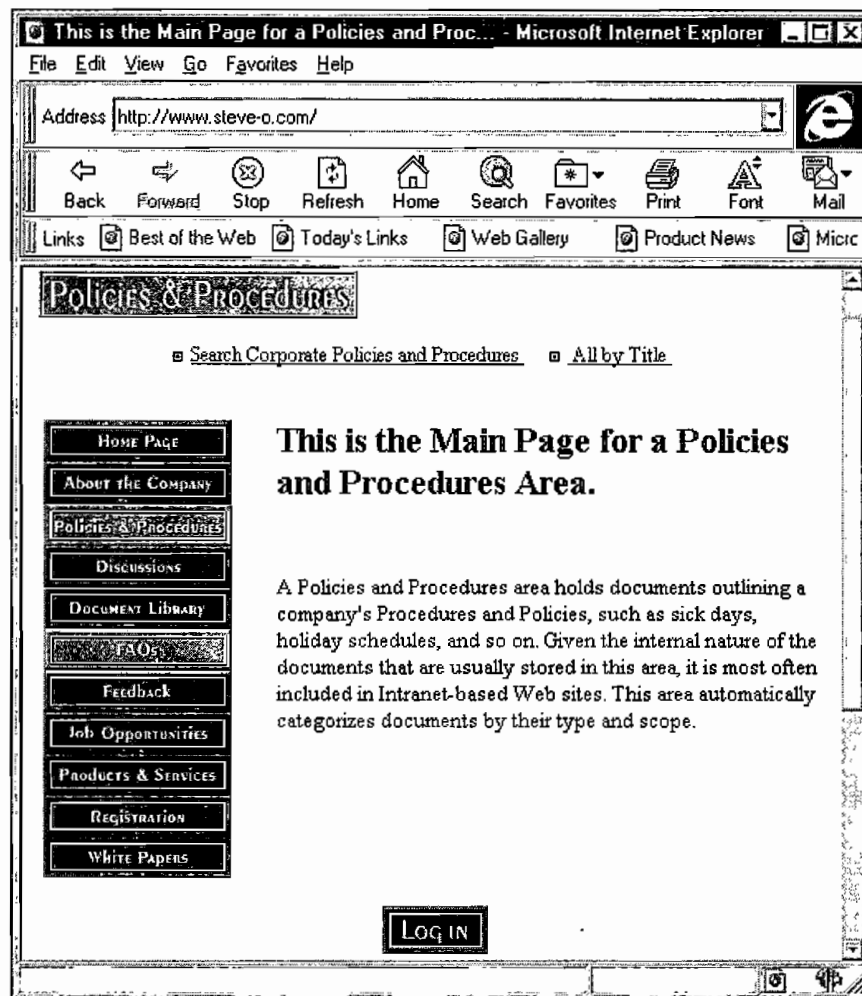


**Figure 26.3** The Main page of the Policies and Procedures area on a sample Domino.Action Web site.

4. **Discussion** offers the site's users the chance to participate in a forum to ask questions, seek customer service, or (hopefully, not) voice complaints. This database area is useful in both Intranet and Internet sites. It includes an approval process so that the Domino.Master can approve the publication of the pages that have been submitted by Web users.



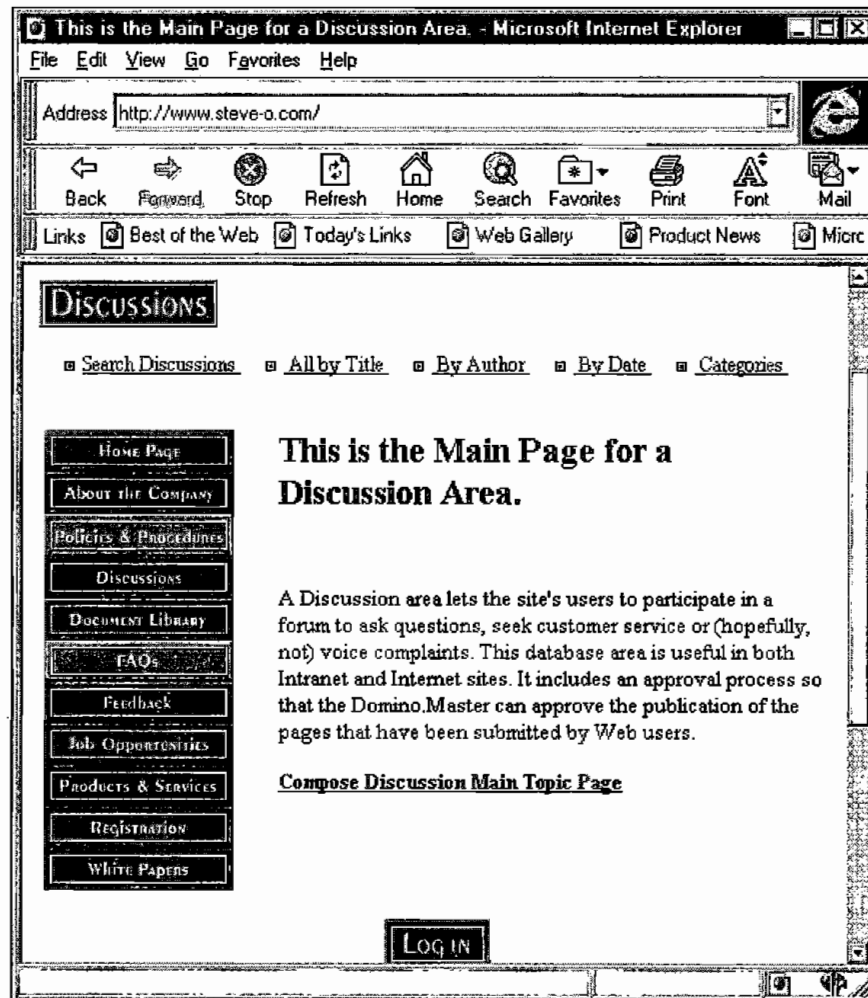


**Figure 26.4** The Main page of the Discussion area on a sample Domino.Action Web site.

5. **Document Library** categorizes the documents in it up to five different ways. This Domino.Action area is similar to the FAQ area, and is often used to hold documentation and reference material, contracts and other legal documents, financial statements and annual reports, blueprints, ad campaigns, and so on.

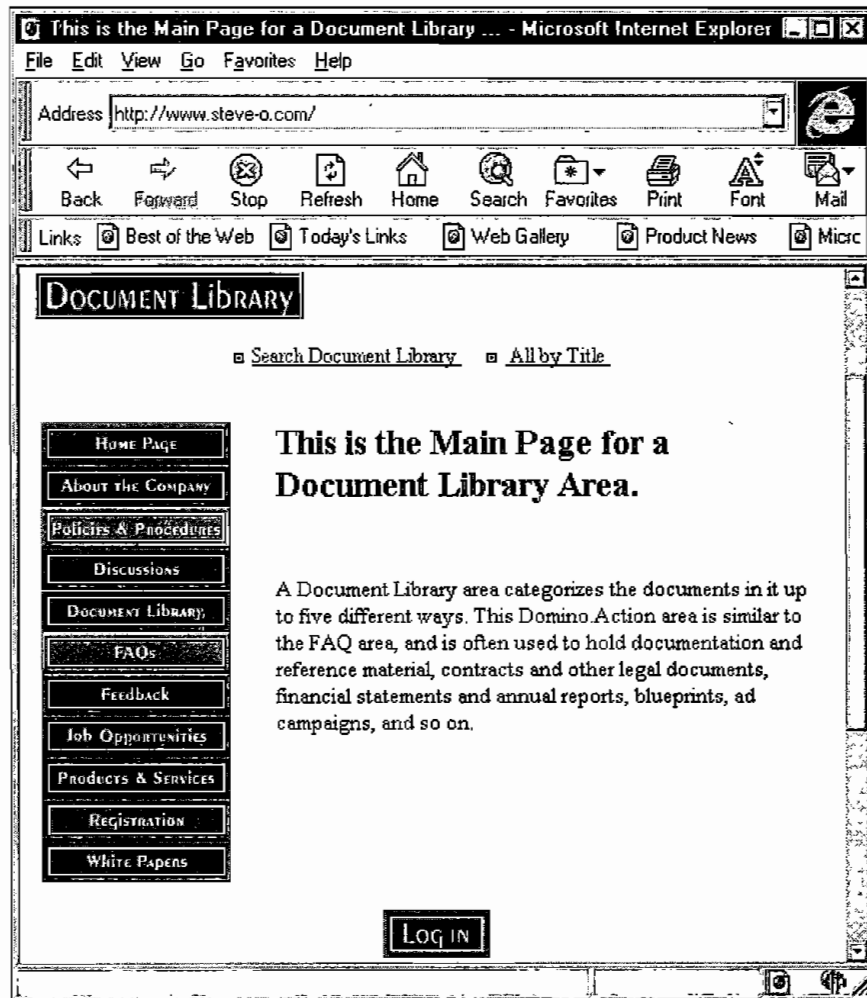


**Figure 26.5** The Main page of the Document Library area on a sample Domino.Action Web site.

6. **Feedback** lets the Domino.Master define pointed survey-type questions to be put to the site's users. This area is used to collect the users impressions of a company, the documents stored in the Web site, and the design and layout of the area itself. A Feedback area differs from a Discussion area in that it requests more specific information for the user, also based on the fact that a Feedback area is intended to facilitate one-way communication (from users to you).

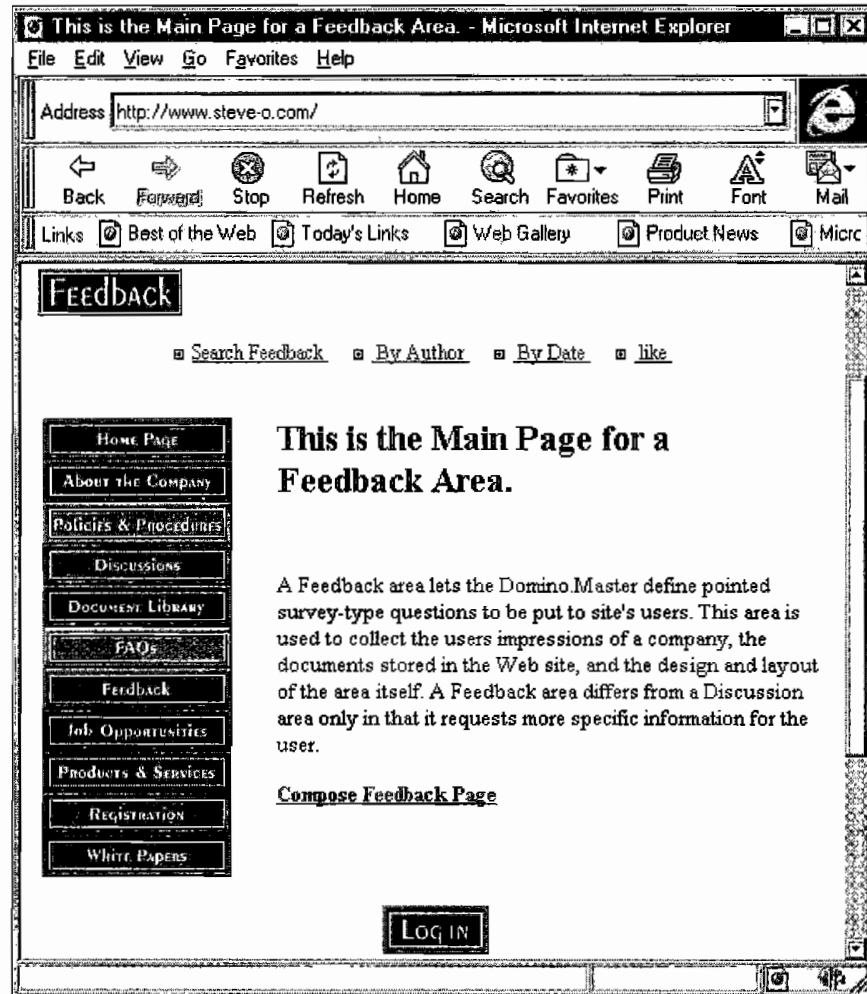


**Figure 26.6** The Main page of the Feedback area on a sample Domino.Action Web site.

7. **Frequently Asked Questions** is similar to the White Paper area, in that it typically holds the answers to a company's or department's most-often asked questions. This area is used by both Intranet and Internet based sites. Documents are organized by category.

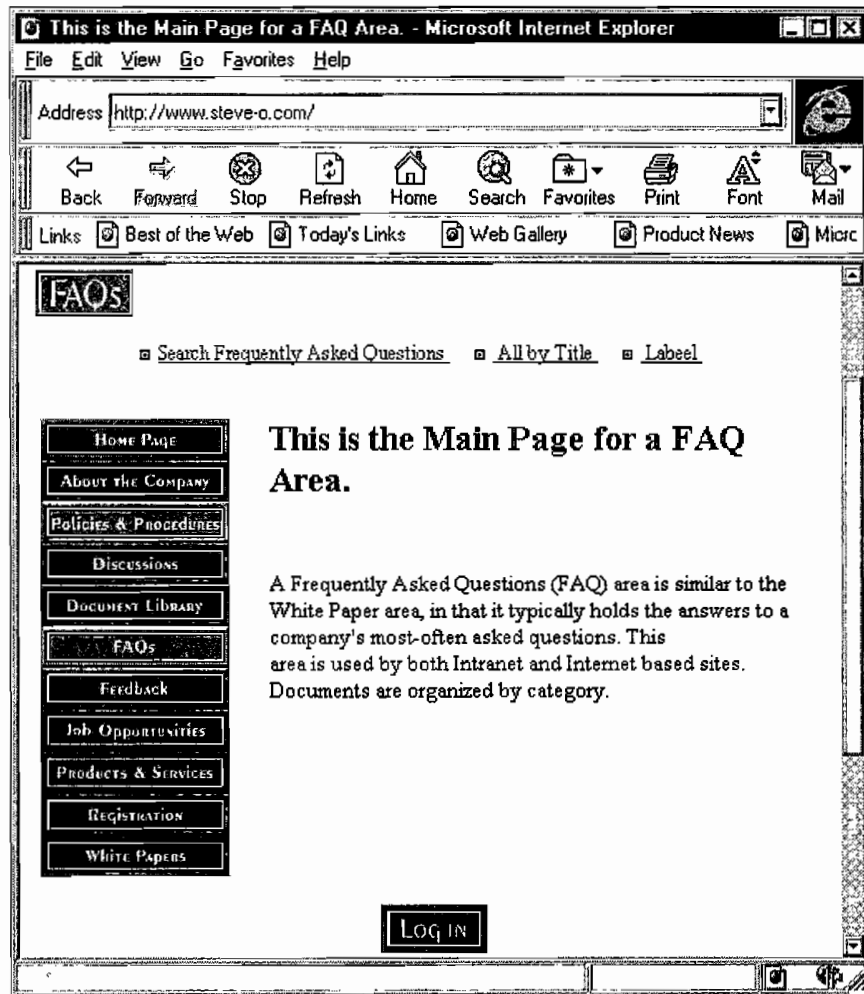


**Figure 26.7** The Main page of the FAQ area on a sample Domino.Action Web site.

8. **Job Postings** lists open employment opportunities in either a department or company, and is useful for both Internets and Intranets. Documents are potentially listed by title, department, by payroll type, or by schedule; in fact, you can define up to five customized categories for documents in this area.

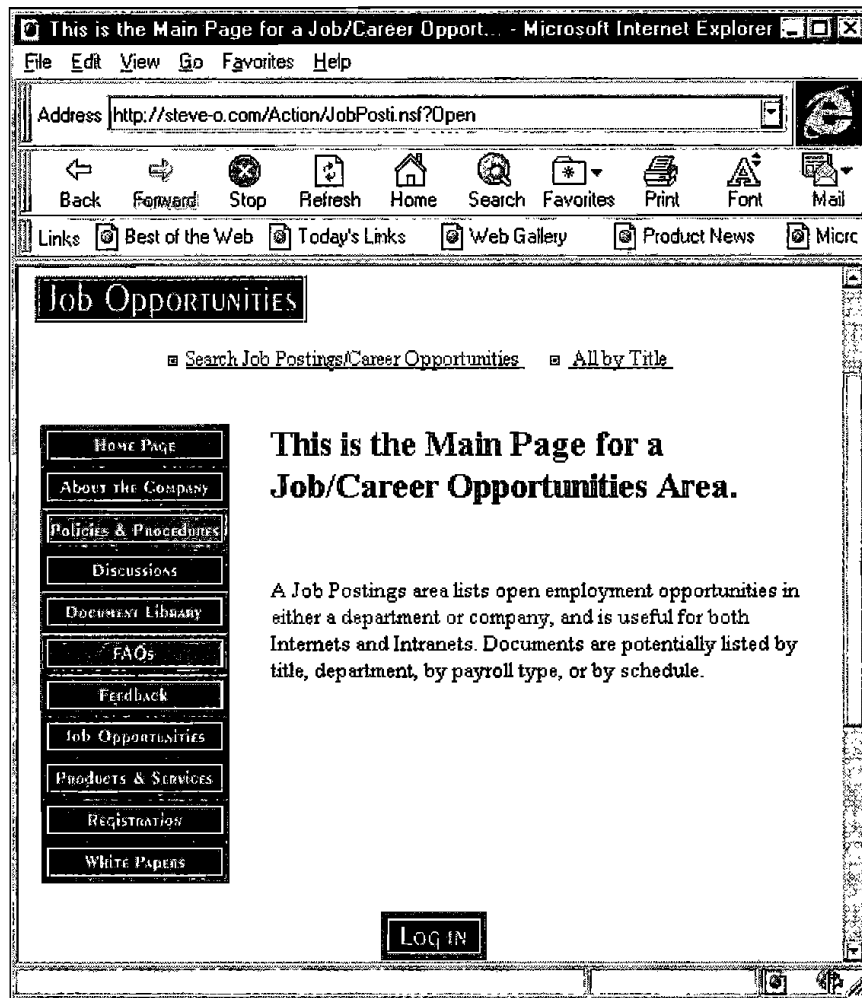


**Figure 26.8** The Main page of the Job Postings area on a sample Domino.Action Web site.

9. **Products and Services** stores documents that describe a company's wares, and is most often used in Internet-based sites. Each of the products included in this area can be described by three pages: a Primary Description page, a Product Specification page, and a Product Review page. The pages in this area can also be organized in up to five categories.

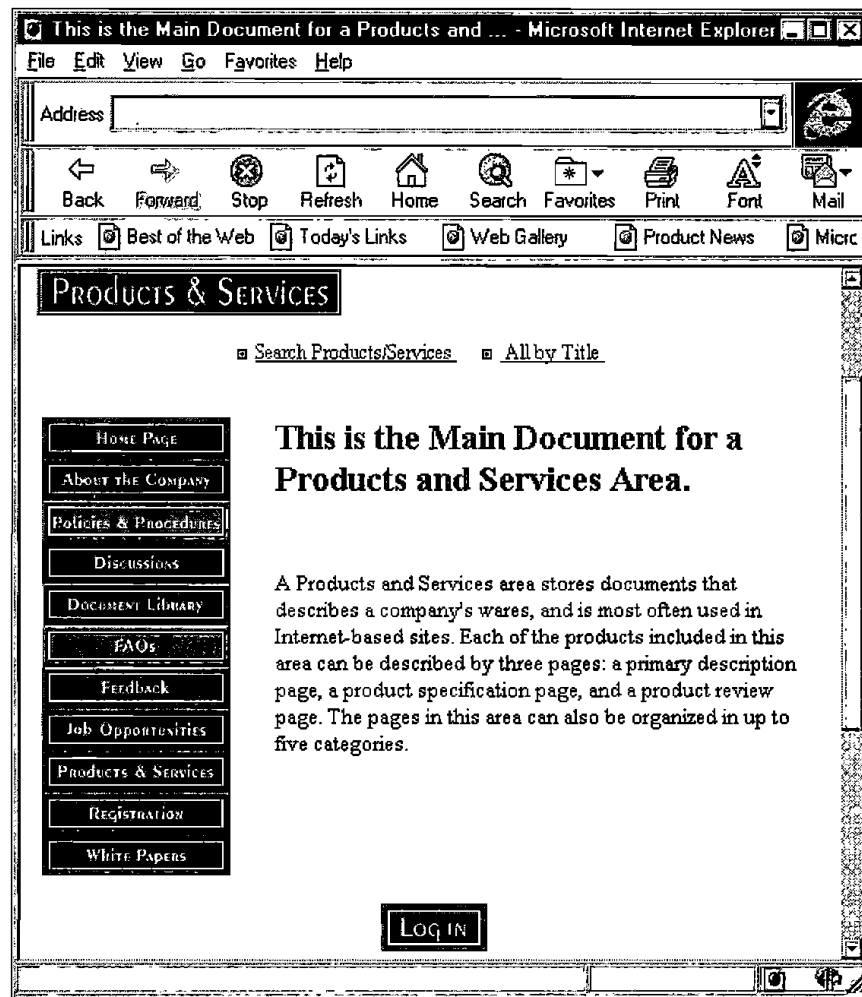


**Figure 26.9** The Main page of the Products and Services area on a sample Domino.Action Web site.

10. **Registration** is intended to let Web users create User Names and Passwords for themselves, and comes with built-in Agents that add new user requests to the N&A Book and appropriate groups. It will also automatically send e-mail to new users, and allows for survey questions along with the User Names and Password.

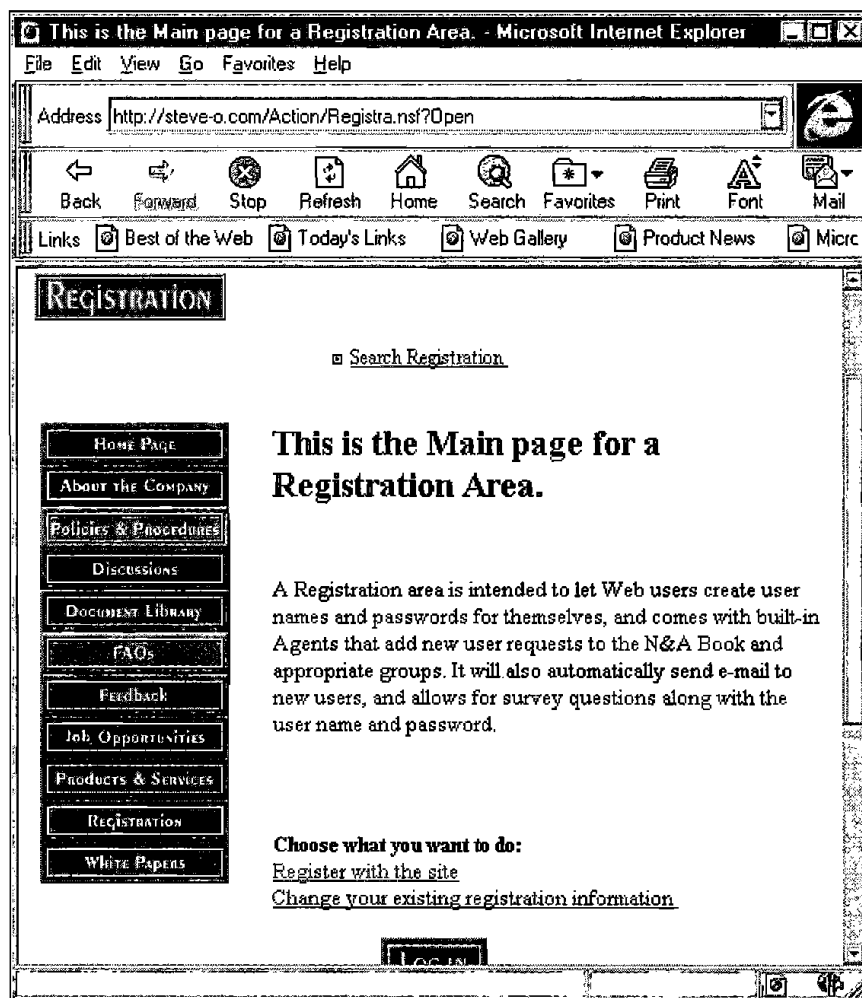


**Figure 26.10** The Main page of the Registration area on a sample Domino.Action Web site.

11. **White Paper** holds documents, such as technical and strategic documents, press releases, corporate communiqués, and white papers. It is similar to the Document Library section, and is useful for both Internets and Intranets.

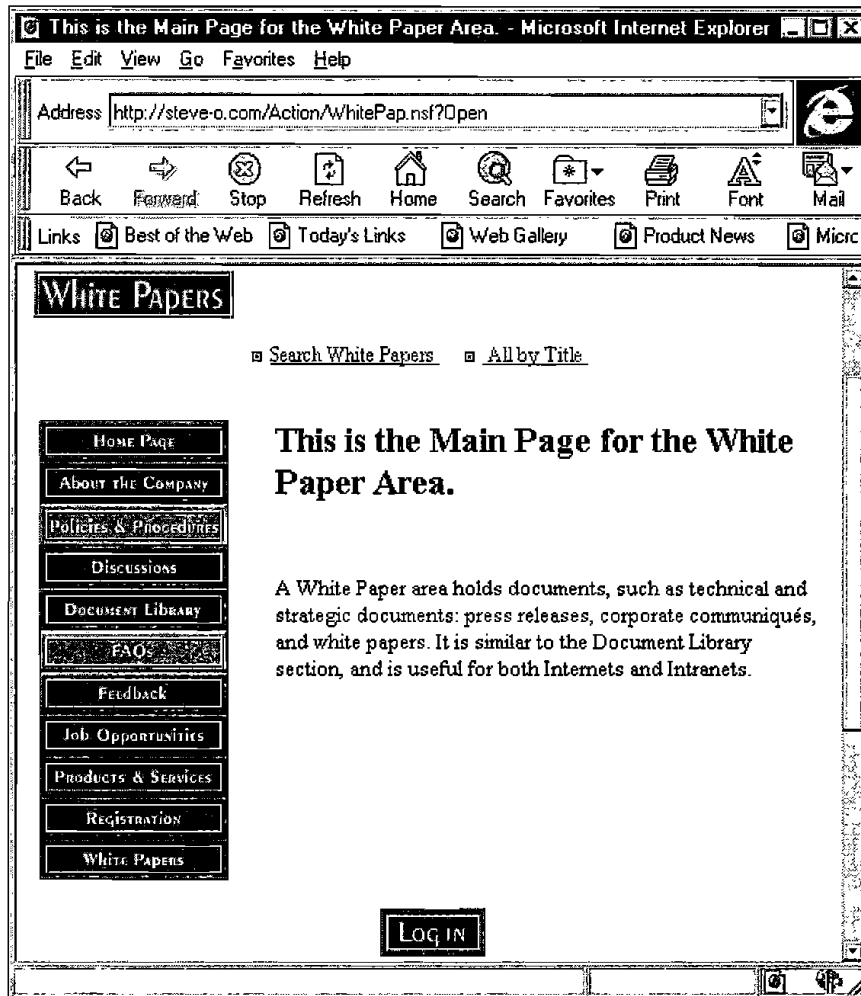


**Figure 26.11** The Main page of the White Paper area on a sample Domino.Action Web site.

## Workflow and Security

One of the most important issues facing the “Domino.Webmaster” concerns the security and workflow model that will be used throughout the areas of the site. Specifically, the documents in each area can be put through a fairly rigorous approval process between the time they are submitted to the database and when they are available on the Web, or the databases making up the various areas of a site can be configured such that documents do not require any approval at all, and are available on the Web as soon as they are added to the database.



Most sites use a combination of security and workflow models, depending on the areas that make up the site; for example, the approval process for documents in a Corporate Policies and Procedures area would certainly be more stringent than that in, say, the site's Discussion area.

In general, the people who interact with a Domino.Action site can be divided into four distinct camps:

1. **Webmaster**, who is in charge of the entire site's management and design;
2. **ContentComposers**, who compose and approve the informational documents in the various areas of the site;
3. **Registered Users**, who typically have access to all of the areas in a site, and may have Author access to some;
4. **Non-registered Users**, who usually have limited access to some of a site's areas, and no access at all to others.

Each area or database on a Domino.Action site requires the Webmaster to make two important decisions: namely, he (or she) must determine (1) what level of Access each type of user has, and (2) the number of approvals (if any) that must be collected before a new document is made available on the Web. Table 26.1 shows a typical site's workflow and security.

**Table 12.1** Security Recommendations

| Site Area               | Approvals needed? | Access for the Webmaster | Access for the Content Providers | Access for Registered users | Access for non-registered users |
|-------------------------|-------------------|--------------------------|----------------------------------|-----------------------------|---------------------------------|
| Home Page               | Usually           | Manager                  | Editor                           | Reader                      | Reader                          |
| Policies and Procedures | Almost Always     | Manager                  | Editor                           | Reader/No Access            | No Access                       |
| Discussion              | No - usually      | Manager                  | Editor                           | Author                      | Reader                          |
| Feedback                | No                | Manager                  | Editor                           | Author                      | Author                          |
| FAQs                    | Usually           | Manager                  | Editor                           | Reader/No Access            | No Access                       |
| Job Postings            | Almost always     | Manager                  | Editor                           | Reader/No Access            | No Access                       |
| Products & Services     | Usually           | Manager                  | Editor                           | Reader/No Access            | Reader                          |
| Registration            | No                | Manager                  | Editor                           | Author                      | Author                          |
| White Paper             | Usually           | Manager                  | Editor                           | Reader/No Access            | Reader/No Access                |

## USING DOMINO.ACTION TO CREATE A WEB SITE

Lotus says there are three steps to creating a new site, but there are actually five:

1. Creating new Domino.Action Library and SiteCreator databases.
2. Configuring the Site
3. Designing the Site
4. Generating the Site
5. Verifying the Site

### Creating New Library and SiteCreator Databases for Domino.Action

In most cases, the database templates that are needed to create the new Library and SiteCreator are located in the Domino server's data directory. If they are not, ensure that they were installed when the Lotus Notes 4.5 software was installed on the Domino server.

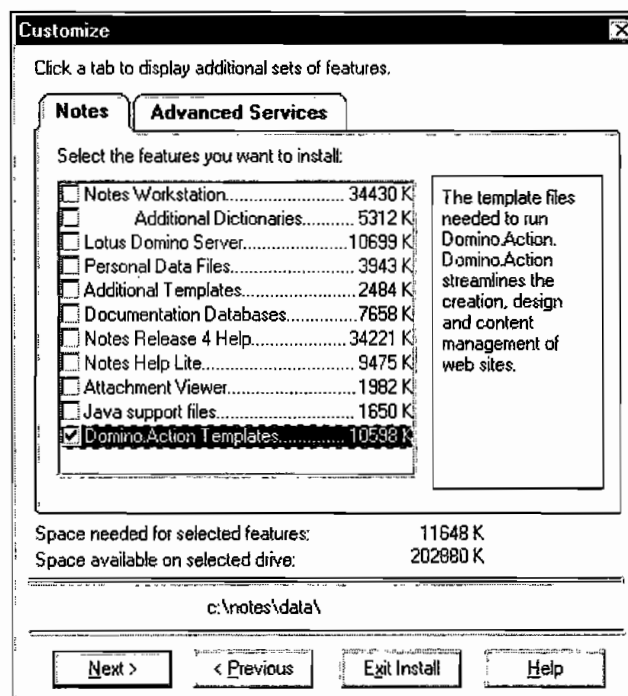


Figure 26.12 Installing the Domino.Action databases.

To create a new Library database—the database that holds all of the design elements that will be copied by the SiteCreator software into the site's databases—start the Notes workstation, and choose **File+Database+New**. Enter the new database's **Title** and **File** name in the New Database dialog box, and choose **Domino.Action Library** from the Template list. (Although the database can have any title and file name, it is best to use `LibAct.nsf` as the file name.)

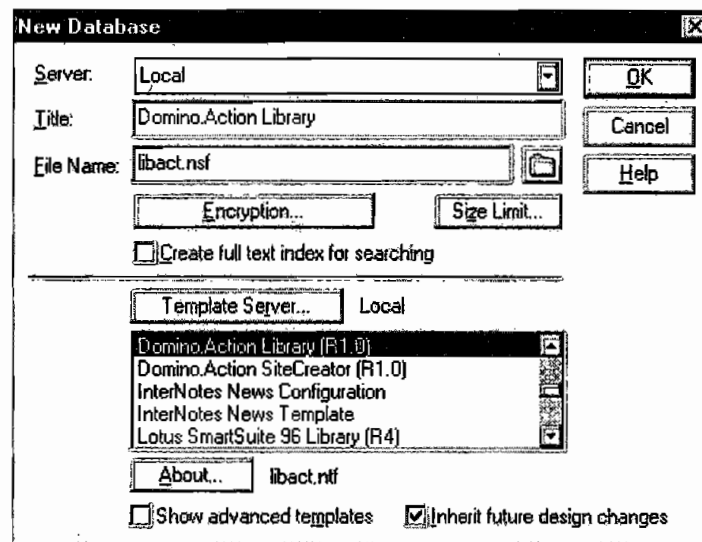


**Figure 26.13** Creating a new Library database for Domino.Action.

To create a new SiteCreator database—the database that contains the Forms and agents that actually do the work of creating the site—choose **File+Database+New**. Enter the new database's **Title** and **File** name in the New Database dialog box, and choose **Domino.Action Site Creator** from the Template list. (Although the database can have any title and file name, it is best to use `SiteAct.nsf` as the SiteCreator database's File Name.)

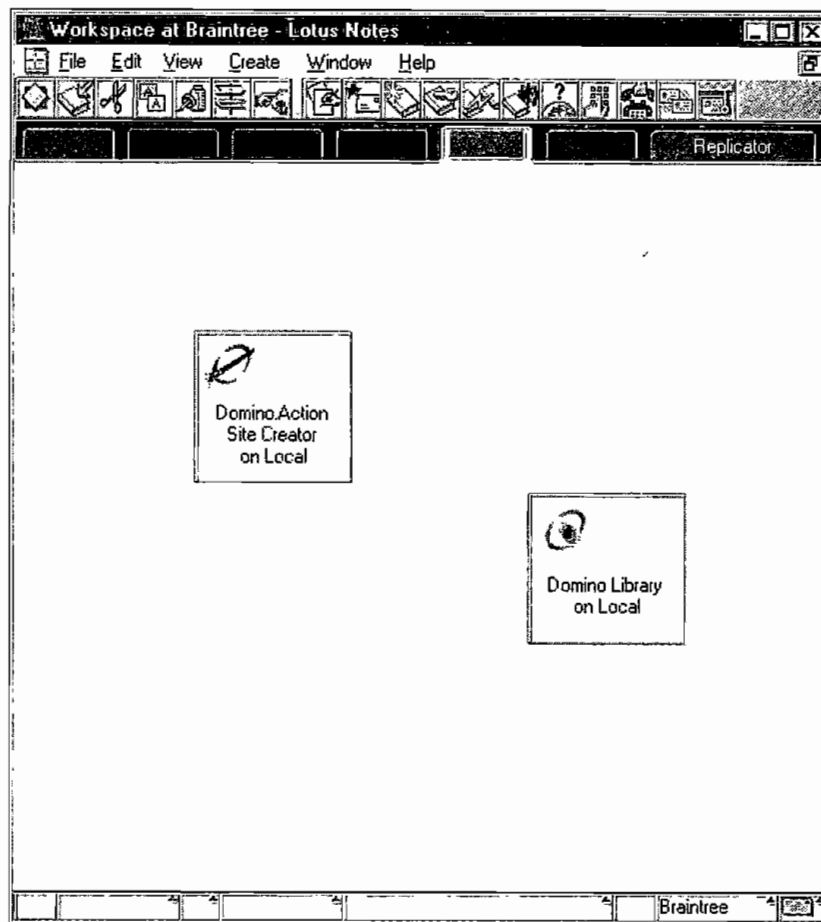


Figure 26.14 The Domino.Action SiteCreator and Library databases.

## CONFIGURING YOUR SITE

All of the work takes place in the SiteCreator database; you do not interact directly with the Library database. Figure 26.15 shows the SiteCreator database, which is divided into three parts: a window listing the Steps involved in creating a new site, a window showing details/asking questions about a particular Step, and the database's title.

To start configuring a new Domino.Action site, choose **Step 1:** in the Steps window, and then click the **Configure Your Site** button in the Form window. Click the **Edit** button at the top of the Form to edit the document and answer the questions about how the site is to be configured.

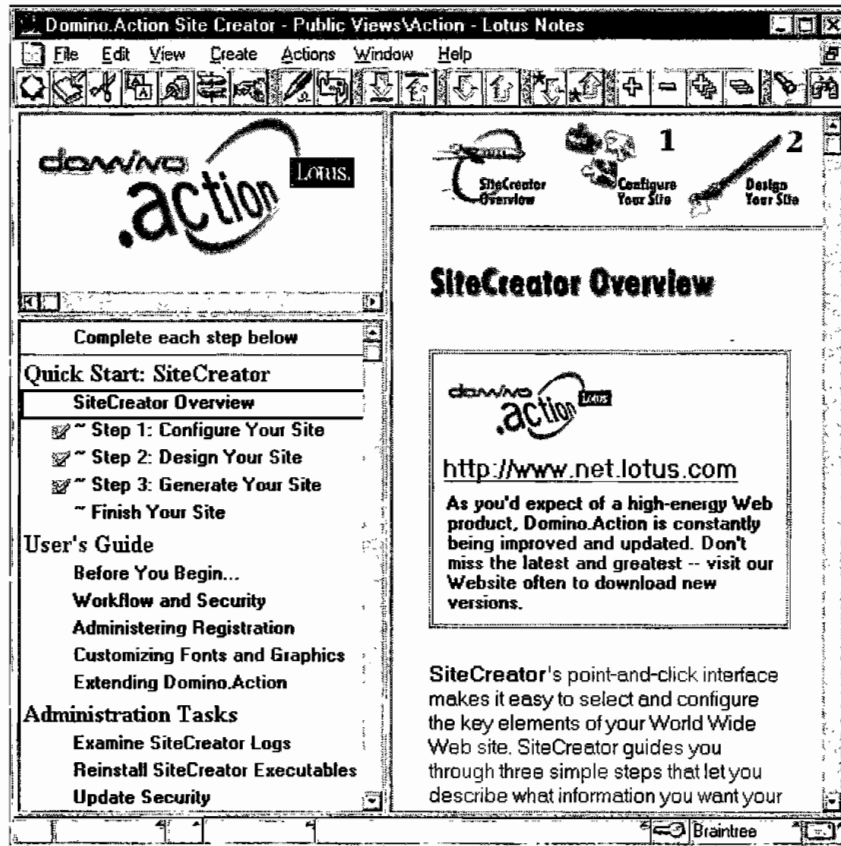


**Figure 26.15** The Domino.Action SiteCreator.

Perhaps the most important part of this first step is the **Site Information** section of the **Configuration Form**—specifically, the **Site Areas** check boxes, which are used to determine which areas the site will contain, which databases the SiteCreator will create, and which areas you will be asked to design and configure. Check off each area that will make up the site. (You can change your mind later and add or remove areas from the site.)

The **Site Managers** field is used to gather the names of the people, servers, and groups who will have Manager access to all of the areas and databases on the site.



**NOTE**

Enter the name of a registered Notes workstation user twice: once in its fully hierarchical format (**Stephen Londergan/NE/Braintree**) and its common format (**Stephen Londergan**).

The values entered in the `Company Name`, `Default Copyright`, `Company Contact Information`, and `Company Contact E-Mail Address` fields are used throughout the pages of the entire site. For example, although any one page's layout can be configured to use a special Copyright message, the default is to display the contents of the `Default Copyright` field on virtually every document in the site.

Use the **Site Images** button to attach any customized graphic files—such as your company or organization's logo, a special background image, or other such picture file—to the database. Whichever file(s) you attach with this button are available for display on any page in the site.

The fields in the `Site Location` section of the Form determine on which server the database files for the site will be created, and in which Directory on the site.

The fields in the `Site Library` section of the Configuration Form tell the SiteCreator where the Library database is located, and what its file name is. (Make sure that the value in the `Library File Name` field matches the name used when the Library database was created.)

Use the **Continue** button (with the blue arrow) at the bottom of the Form to save the document and proceed to Step #2.

## DESIGNING THE SITE

After clicking the **Continue** button, the title screen for Step #2 is shown, as is displayed in Figure 26.16.

As is shown in the figure, the SiteCreator will let you skip the (potentially arduous) process of defining the options for each of the areas in the site by clicking the **Quick Design** button. Doing so makes the SiteCreator apply the simplest layouts and approval processes to each of the documents in each of the areas you chose to create in Step #1. (If you so choose the easy way out, you should at least review the Security settings for each of the documents and areas.)

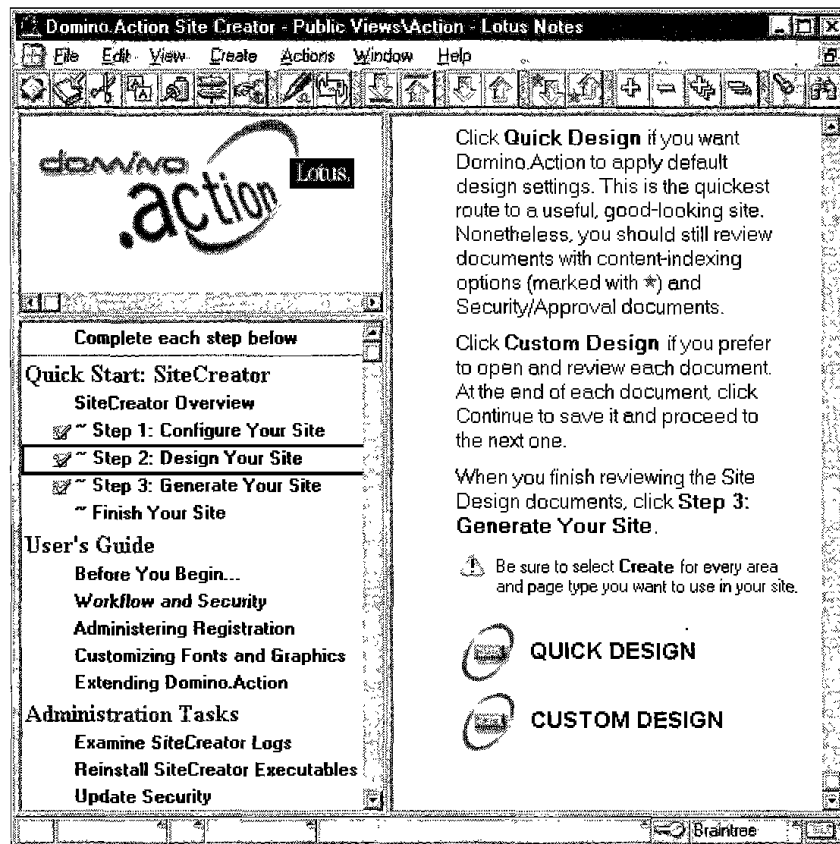


Figure 26.16 Step #2.

In most cases, you will click the **Custom Design** button to individually configure and format the areas and pages that will make up the Web site. The number of questions you'll have to answer (and the amount of time it will take you to complete them) depends entirely on how many areas you chose to include in the site in Step #1.

The rest of this section outlines each step in greater detail. Many of the areas use the same basic layout and steps, so the following sections are intended as a reference for your use when defining each type of area.

## Designing a Registration Area

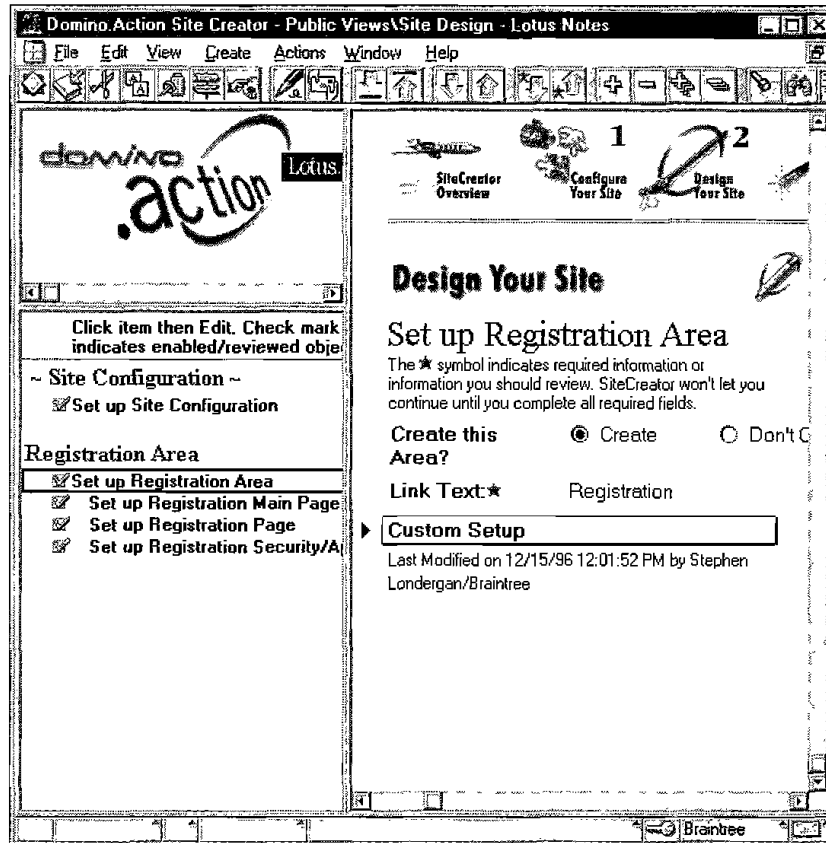


**Figure 26.17** Designing a Registration Area

There are four parts to designing a Registration area: setting up the area itself, setting up its main page, setting up the Registration page, and determining the Security and Approval settings for the area.

Use the document list at the left of the screen to select which of the four steps you intend to complete, and then use the **Edit** button in the document window on the right to put the document into Edit mode and answer the questions.

### Setting up the Registration Area

The first document has only three fields.



Use the `Create this area` radio buttons to determine whether or not the SiteCreator will assemble this section of the site. (You would set this field to **Don't Create** only when modifying an a site that had already been created; always set this field to **Create** when designing the area for the first time.)

The `Link Text` field is used to determine how Links to this area will be represented in other areas of the site.

Use the `Area Database File Name` field (in the `Custom Setup` section of the Form) if you need to change the area's default file name of `Registra.nsf`.

#### Setting up the Registration Main Page

The next page is used to set up the Main page for the Registration area in other words, the page that is displayed to users who open this part of the site. Similar to the previous page that is used to set up the entire area, the first field is used to determine whether the SiteCreator will assemble this page; in most cases, leave the `Create this Page` field set to the default value of `Create`.

The `Page Look` field is used to determine the overall style and layout of the area's Home Page. Use the first **Click here** button to choose the page's Layout and Link style; use the second **Click here** button to preview what the page will look like.



#### NOTE

---

In most cases, the Page Look you choose for the Registration area should be the same as that you choose (or plan to choose) for the other areas of the site.)

---

Clicking the **Background Color** button produces a palette from which you can choose a color that will be used for the background of the Registration area's Home Page.

In addition to the `Background Color` field, you can use the `Background Image` field to choose a Background picture that will be used for the page. Use the first **Click here** button to choose the image; use the second **Click here** button to preview the page with the chosen background image.



In addition to the default Basic Bottom, Basic Side, Basic Top, and None options, the Background images available will include any graphic files that were included in the site in Step #1.

The Custom Setup area of the Form can be used to override the default copyright message that will be displayed on the page.

#### Setting up the Registration Page

The next page in the Registration area is used to set up the page that is displayed to users who are registering at the site. The first fields on the Form—Create this Page, Background Color, and Background Image—function the same as the similarly-named fields on the previous document.

The Registered Users Group is used to store the name of the group in the Public N&A Book on the Domino server to which new registered users will be added. This group is the linchpin of the site's security, and it should be used in all of the database's Access Control Lists to determine what level of access is assigned to registered users of the site.

The Public Address Book field is normally left with the default value of NAMES.NSF; if, for some reason, the Public N&A Book on the Domino server has a different name, enter it in this field.

The Internet Gateway Address field holds the name of the domain through which mail destined for the registered user should be sent. (When a new user registers, an agent in the Registration database sends the user an e-mail message confirming the registration. This field is used by the agent to determine the convention for addressing mail to a person via an Internet Gateway.)

The Personal Information field determines whether users registering for the site are prompted for some additional information, such as their company name, address, and so on. (Even if this field is checked, users have the option of leaving these fields blank when registering.)

The Custom Questions field lets you define up to five additional questions that will be put to people when they fill out a Registration Form. If you choose to ask for additional information, enter the text of the custom question(s) in the Question 1, Question 2, etc. fields, and

the possible answers—separated by semicolons—in the Answers 1, Answers 2, etc. fields. Use the **Allow Other** button to determine whether or not Web users can enter a response other than what you have specified in the Answers 1 fields.

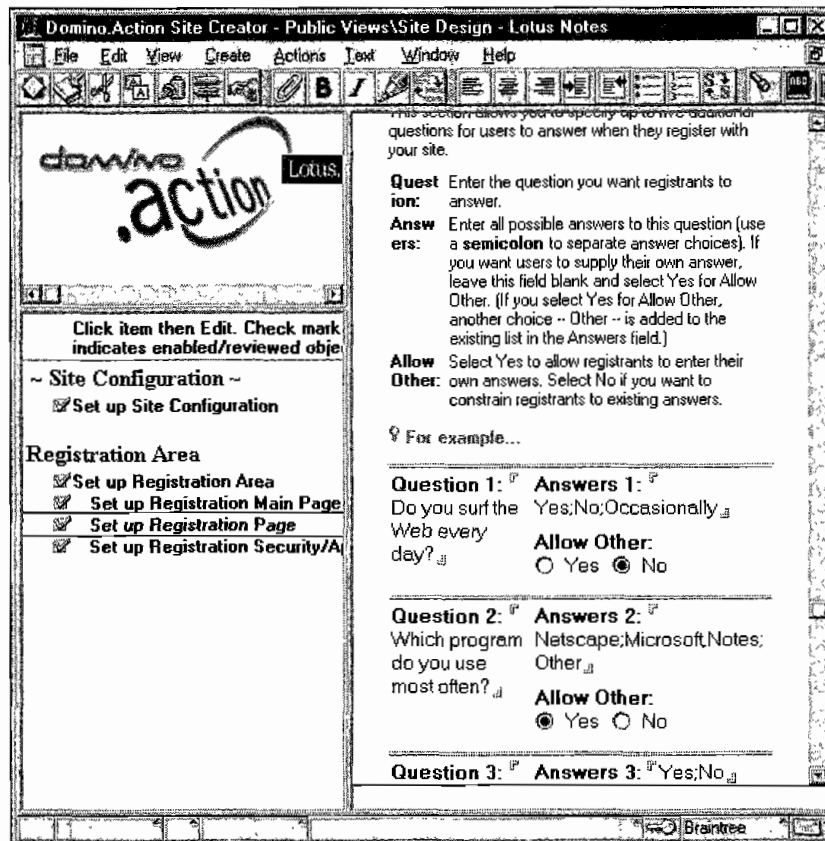


**Figure 26.18** You can ask your own customized questions of users who are registering on the site.

As with the previous Form that was used to configure the area's main page, use the Display Copyright field in the Custom section to alter the default copyright message that will be displayed on the Web page that people use to register with the site.

### Setting up the Registration Security and Approvals

The next page in the Registration Area configures the security and workflow settings for this area of the Web site.

The `Content Composers` field stores the names of the people and groups who can add documents to this part of the site.

The default value for the `Allow anonymous access` field in the Registration area is set to **Yes**, and this may be the one part of the site (besides the site's Home Page) where **Yes** is a legitimate answer for this field. Since everyone who uses the Registration area will be an anonymous user at first—that is, after all, why the user would want to register in the first place—this field is usually set to **Yes**. Setting this field to **No** will prevent unregistered users from using the Registration area of the site.

Similarly, and for the same reasons, the `Allow anyone to author` field is usually also set to **Yes** in this area of the site. Setting this field to **No** prevents un-authenticated/unregistered users from composing documents in this area, and reserves the right to author/edit/delete documents for the people listed in the `Content Composers` field.

If the `Number of approval steps` field is set to any value other than zero, the `Automatic Notification` field appears. Use this field to configure the database's agents to automatically send a notification e-mail message to a document's author and approvers whenever it is edited. (The Domino server must be configured and able to route Notes-based e-mail for this to work.)

The `Approvers Names and Functions` section of the document tell the database the virtual "chain of command" of names who must approve a document before it will be displayed on the Web. Use the optional `Function` field to enter a short description of why the person is listed as an approver. (See Chapter 27 for a detailed discussion of the Domino.Action workflow processes.)

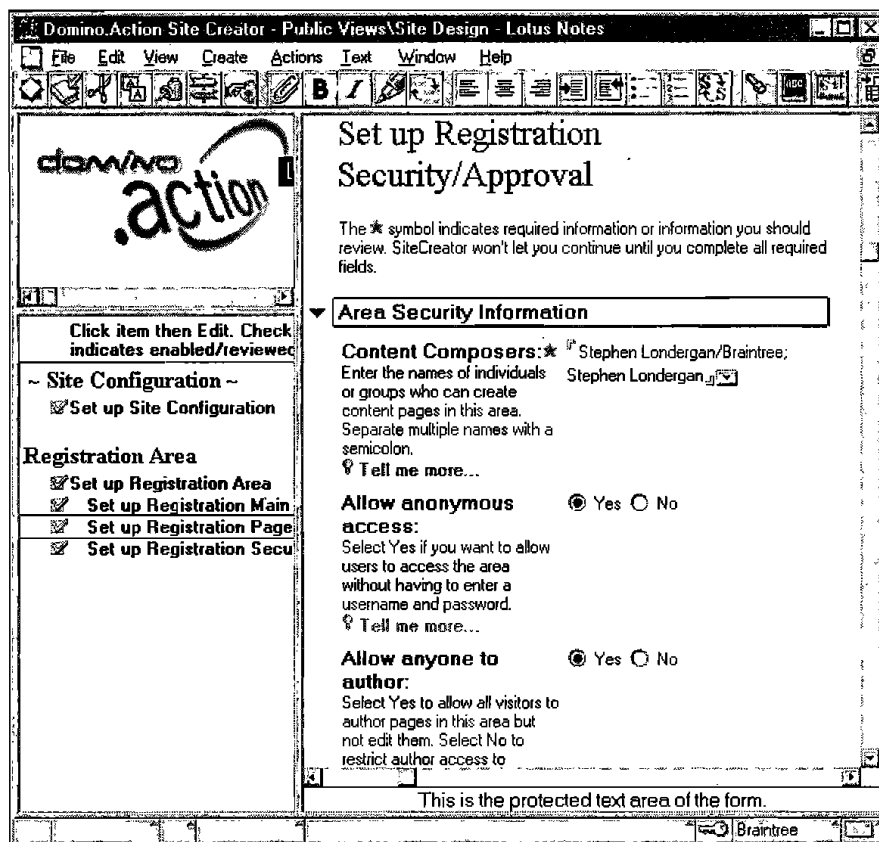


**Figure 26.19** Defining the people who must approve a new document before it will be available on the Web.

## Designing a Home Page Area

Just as with the Registration area, there are four parts to designing a Home Page area: setting up the area itself, setting up its Main page (which will be the Home Page for the entire site), setting up the General Information page, and determining the Security and approval settings for the area. The good news is that the steps involved in setting up a Home Page are almost identical to those described in the section about setting up the Registration area.

### Setting up the Home Page Area

The first document has only three fields; since they function exactly the same way as those that were described in the previous section, the descriptions that follow are understandably short:

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.
- The `Link Text` field determines how Links to this area will be represented in other areas of the site.
- The `Area Database File Name` field (in the `Custom Setup` section of the Form) can be used to change the area's default file name of `HomePage.nsf`.

### Setting up the Home Page

The next page is used to set up the Main page for the Home Page area—this will be the page that usually serves as the Home Page for the entire site. By default, it will have Links to all of the other areas on the site, as well as Links to other documents within the Home area itself. Other than its purpose, setting up this page is very much like setting up the Main page for the Registration area:

- The `Create this Page` field determines whether the SiteCreator will assemble this page; this field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the site's Home Page. The Page Look you choose for the Home Page should match the Look you choose (or plan to choose) for other areas of the site.)
- The `Background Color` button determines the Home Page's background color.
- The `Background Image` field determines which (if any) image will be used as the Home Page's background. The choices available will be based on which images were included in Step #1.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

### Setting up the General Information Page

In addition to serving up the site's Home Page, the Home Page area/database might also contain content documents its own—such as directions to the company, hours that the store is open, and so on. Use the next Form in the SiteCreator database to determine this page's layout and color.

- The `Create this Page` field determines whether the SiteCreator will assemble this page; this field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` can be used to override the default copyright message.

### Setting up the Home Page Area's Security and Approvals

Since the Home Page area may contain the previously-described content (directions, hours of operation, etc.), it needs to have its own set of Security and Approval settings. The Security for this area is usually quite different from that of the Registration area, since the only users typically awarded the right to add documents to this part of the site are the Content Creators; Web users do not usually add documents to this part of the site.

- The `Content Composers` field lists the people and groups of people who can add documents to this area of the site.
- The `Readers` field determines who can read the documents in this area. Since a site's Home Page is usually available to anyone, leave this field set to the default value of **-All-**. Alternately, use the field's drop-down list to choose the names of the people who can read the Home Page area's documents.

- The Number of approval steps section of the Form functions exactly the same as the Registration area's; if the Number of approval steps field is set to any value other than **zero**, the Automatic Notification field is used to configure the database's agents to automatically send a notification e-mail message to a document's author and approvers whenever it is edited. The Approvers Names and Functions section of the document tells the database the virtual "chain of command" of names who must approve a document before it will be displayed on the Web.

### Designing an "About the Company" Area

Just as with the Registration and Home Page areas, there are four parts to designing an About the Company area of a Domino.Action site: setting up the area itself, setting up its Main page, setting up the About the Company page, and determining the Security and Approval settings for the area.

#### Setting up the "About the Company" Area

- The Create this area radio buttons determine whether or not the SiteCreator will assemble this section of the site.
- The Link Text field determines how Links to this area will be represented in other areas of the site.
- The Area Database File Name field (in the Custom Setup section of the Form) can be used to change the area's default file name of About.nsf.

#### Setting up the "About the Company" Main Page

The next Form configures the appearance and layout of the Main page for this area of the site.

- The Create this Page field determines whether the SiteCreator will assemble this page. Like the others, this field should usually be left with the default value of **Create**.



- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

#### Setting up the "About the Company" Page

The "About the Company" page is used to display any of the area's content documents—in other words, the press releases, corporate background documents, and so on—that have been added to the database by the Content Composers.

- The `Create this Page` field determines whether the SiteCreator will assemble this page; this field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` can be used to override the default copyright message.

This Form has one special section that distinguishes it from the other areas of the database: the `Content Index` section of the Form is used to set up Categories that the people adding documents to the database can use to organize their submissions.

Enter the type of category in the `Label` fields, and the possible values for the category type in the `Choices` fields. Use the `Allow Other` fields to specify whether or not contributors can categorize documents in categories other than those that you have defined in the `Choices` fields.

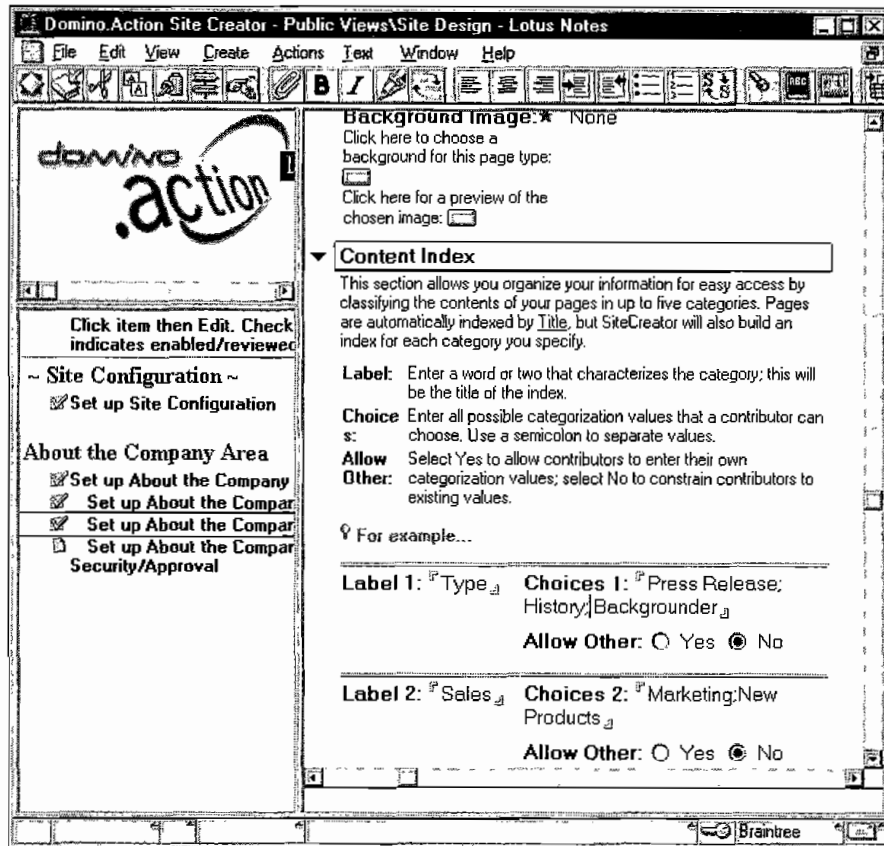


**Figure 26.20** Defining the categories that can be used to organize the documents in the About the Company area of the site.

### Setting up the “About the Company’s” Security and Approvals

Like the Home Page area of most sites, an “About the Company” area is typically allows only the site’s Content Composers to add documents to the area; in fact, this area may be one in which the Readers are limited to registered users of the site only. If that is the case, use the Readers field to choose the people and groups who have access to this area of the site.



**NOTE**

Make sure to add the group name you chose for registered users in the Registration area to the About the Company database’s Access Control List if you want to limit access to this area.

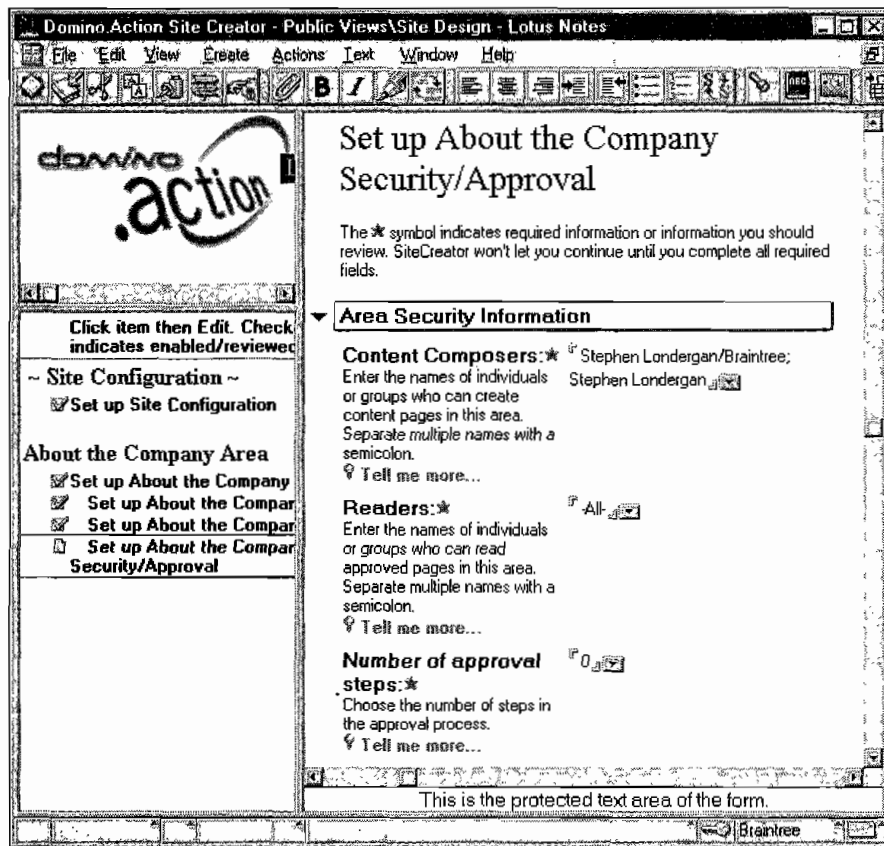


**Figure 26.21** Limiting the access to an area.

- The **Content Composers** field lists the people and groups of people who can add documents to this area of the site.
- The **Readers** field determines who can read the documents in this area.
- The **Number of approval steps** section of the Form functions exactly the same as the Registration and Home Page areas.

## Designing a Corporate Policies and Procedures Area

There are four parts to designing a Corporate Policies and Procedures area: setting up the area itself, setting up its Main page, setting up the Corporate Policies and Procedures page, and determining the Security and Approval settings for the area.

### Setting up the Corporate Policies and Procedures Area

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.
- The `Link Text` field determines how Links to this area will be represented in other areas of the site. (This is one area that has a particularly long name, which will produce particularly long Links in the other areas of the site, so you might consider changing it in favor of something shorter.)
- The `Custom Setup` section of the Form can be used to change the area's default file name of `CorpPoli.nsf`.

### Setting up the Corporate Policies and Procedures Main Page

The next Form configures the appearance and layout of the Main page for this area of the site, which is used as the area's Home Page.

- The `Create this Page` field determines whether the SiteCreator will assemble this page. Leave this field with the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

### Setting up the Corporate Policies and Procedures Page

When a contributor adds a document to the database describing a new policy or procedure—like the number of sick days, or perhaps a holiday schedule—it will be displayed on the Web through this page style.

- The `Create this Page` field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` can be used to override the default copyright message.

Like the "About the Company" Form, this Form has a `Content Index` section that is used to define up to five categories that the people adding documents to the database can use to organize their submissions.

Enter the type of category in the `Label` fields, and the possible values for the category type in the `Choices` fields. For example, creating a set of categories for a Human Resources application might include **New Policies** in the `Label` field, and **Sick days;Holiday Time Off;Vacation Carryover** in the `Choices` field (separate multiple choices with semicolons). Use the `Allow Other` fields to determine if contributors can categorize documents in categories other than those that you have defined in the `Choices` fields.

#### Setting up the Corporate Policies and Procedures Security and Approvals

The Security and Workflow settings Form for this are functions just like those that have already been described. This area is normally be set so that only the Content Composers can add documents.

- The `Content Composers` field lists the people and groups of people who can add documents to this area of the site.
- The `Readers` field determines who can read the documents in this area.
- The `Number of approval steps` section of the Form functions exactly the same as the Registration and Home Page areas.

## Designing a Discussion Area

There are five parts to a Discussion area: the area itself, the area's Main page, its Main document Page, the Response page, and the Security and Approval settings.

### Setting up the Discussion Area

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.
- The `Link Text` field determines how Links to this area will be represented in other areas.
- The `Custom Setup` section of the Form can be used to change the area's default file name of `Discuss.nsf`.

### Setting up the Discussion Main Page

The next Form configures the appearance and layout of the Home Page for the Discussion area.

- The `Create this Page` should be left set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

### Setting up the Main Topic Page

When a contributor or user adds a document to the database, it will be displayed on the Web through this page style.

- The `Create this Page` field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's Background.
- The `Banner` field determines which (if any) graphic will be placed at the top of each document composed with the Main topic Form. Use the first **Click here** button to choose the image to be used, and use the second **Click here** button to see what the page will look like with the chosen image.
- The *Custom Setup* can be used to override the default copyright message.

The Discussion area will automatically come with Views that sort the documents by their author, date, or title. Use the optional `Content Index` section to define an additional way that users and Content Contributors can organize their submissions.

#### Setting up the Response Page

Given the "free-wheeling" discourse that takes place in a Discussion area, the database comes pre-made with an additional Form that is used to respond to the documents that other people have composed.

- The `Create this Page` field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.

- The `Banner` field determines which (if any) graphic will be placed at the top of each document composed with the Response Form. Use the first **Click here** button to choose the image to be used, and the second **Click here** button to see what the page will look like.
- The `Custom Setup` can be used to override the default copyright message.

#### Setting up the Discussion Area's Security and Approvals

The Security and Workflow settings Form for a Discussion area are configured much the same as those that have been described in earlier sections of this chapter.

- The `Content Composers` field lists the people and groups of people who can add content documents to this area of the site.
- The `Allow Anonymous Access` field determines if a user can enter a document in the discussion area without a User Name and Password.
- The `Allow Anyone to Author` field determines if the ability to add pages to the area is reserved for just the people named in the `Content providers` field, or is open to everyone.
- The `Number of approval steps` section of the Form determines how many approvals are required before a new page in the Discussion area will be available on the Web.

### Designing a Document Library Area

There are four parts to designing a Document Library: setting up the area itself, setting up its Main page, setting up the Document Library page, and determining the Security and approval settings.

#### Setting up the Document Library Area

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.



- The `Link Text` field determines how Links to this area will be represented in other areas of the site.
- The `Custom Setup` section of the Form can be used to change the area's default file name of `DocLibra.nsf`.

#### Setting up the Document Library Main Page

The next Form configures the appearance and layout of the area's Home Page.

- The `Create this Page` field's default value is **Create**.
- The `Page Look` field determines the overall style and layout of the page, and should match the other pages and areas on the site.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

#### Setting up the Document Library Page

When a contributor adds a document to the Document Library area—such as a product catalog, or list of “fax back” documents—it will be displayed on the Web through this page style.

- Leave the `Create this Page` field set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` can be used to override the default copyright message.

This Form has a `Content Index` section that is used to define up to five categories that the people adding documents to the database can use to organize their submissions.

Enter the type of category in the `Label` fields, and the possible values for the category type in the `Choices` fields. Use the `Allow Other` fields to determine if contributors can categorize documents in categories other than those that you have defined in the `Choices` fields.

#### Setting up the Document Library Security and Approvals

The Security and Workflow settings for this area are often configured so that documents in the area can be composed only by the Content Composers, and can be read by only registered users.

- The `Content Composers` field lists the people and groups of people who can add documents to this area of the site.
- The `Readers` field determines who can read the documents in this area.
- The `Number of approval steps` section of the Form details which (and from whom) approvals are required before new Library documents are available on the Web.

### Designing a Feedback Area

There are four parts to a Feedback area: the area itself, its Main page, the Feedback page, and determining the Security and Approval settings for the area.

#### Setting up the Feedback Area

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.
- The `Link Text` field determines how Links to this area will be represented in other areas of the site.

- The `Custom Setup` section of the Form can be used to change the area's default file name of `Feedback.nsf`.

#### Setting up the Feedback Area's Main Page

The next Form configures the appearance and layout of the area's Home Page.

- The `Create this Page` field's default value is **Create**.
- The `Page Look` field determines the overall style and layout of the page, and should match the other pages and areas on the site.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

#### Setting up the Feedback Page

Comments about the company or its Web site are composed and read through the Feedback Form.

- Leave the `Create this Page` field set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Banner` field determines which image will be displayed at the top of each page in the Feedback area.
- The `Custom Setup` can be used to override the default copyright message.

### Setting up the Feedback Area's Security and Approvals

The Security and Workflow settings Form for a Feedback area are usually set so that anyone—even people without a User Name and Password—can contribute documents to the area.

- The `Content Composers` field lists the people and groups of people who can add content documents to this area of the site.
- The `Allow Anonymous Access` field determines if a user can enter a document in the discussion area without a username and password, and is normally set to **Yes** in a Feedback area.
- The `Allow Anyone to Author` field determines if the ability to add pages to the area is reserved for just the people named in the `Content providers` field, or is open to everyone.
- The `Number of approval steps` section of the Form determines how many approvals are required before a new page in the Discussion area will be available on the Web.

### Designing a Frequently Asked Questions Area

Setting up a FAQ area is just like setting up a Document Library area—with one exception: FAQs have only one possible value for the `Content Index`.

### Designing a Job/Career Opportunities Area

A Job/Career Opportunities area is just like setting up a Document Library area.

### Designing a Product/Services Area

There are six parts to a Product/Services area: the area itself, the area's Product Information page, Product Review page, Product Specification page, and the Security and Approval settings.

#### Setting up the Products/Services Area

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.

- The `Link Text` field determines how Links to this area will be represented in other areas.
- The `Custom Setup` section of the Form can be used to change the area's default file name of `Products.nsf`.

#### Setting up the Products/Services Main Page

The next Form configures the appearance and layout of the Home Page for the Products/Services area.

- The `Create this Page` field should be left set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

#### Setting up the Product Information Page

The Product Information page is used to provide the "high-level" description of the item.

- The `Create this Page` field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` can be used to override the default copyright message.

The Product/Services area will automatically sort the documents by their title; use the optional Content Index section to define up to five additional ways that the documents can be organized.

#### Setting up the Product Review Page

The Product Review page is intended to capture descriptions of how a product or service was used, held up under pressure, etc.

- The `Create this Page` field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page. (Note that Product Review pages have different looks than other page types in Domino.Action.)
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` can be used to override the default copyright message.

#### Setting up the Product Specification Page

This page is intended to store specifications and other additional information about a particular product or service.

- The `Create this Page` field should be set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page. (Note that Product Specification pages have different looks than other page types in Domino.Action.)
- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's Background.
- The `Custom Setup` can be used to override the default copyright message.

### Setting up the Product/Service Area's Security and Approvals

The Security and Workflow settings Form for a Product/Services area typically allow only Content providers to submit documents, but allow anyone to read documents.

- The `Content Composers` field lists the people and groups of people who can add content documents to this area of the site.
- The `Readers` field determines who can read the documents in this area.
- The `Number of approval steps` section of the Form determines how many approvals are required before a new page in the Discussion area will be available on the Web.

### Designing a White Paper Area

There are four parts to a White paper area's configuration: the area itself, its Main page, the White Paper page, and the Security and Approval settings.

#### Setting up the White Paper Area

- The `Create this area` radio buttons determine whether or not the SiteCreator will assemble this section of the site.
- The `Link Text` field determines how Links to this area will be represented in other areas of the site.
- The `Custom Setup` section of the Form can be used to change the area's default file name of `DocLibra.nsf`.

#### Setting up the White Paper Main Page

The next Form configures the appearance and layout of the area's Home Page.

- The `Create this Page` field's default value is **Create**.
- The `Page Look` field determines the overall style and layout of the page, and should match the other pages and areas on the site.

- The `Background Color` button determines the page's background color.
- The `Background Image` field determines which (if any) image will be used as the page's background.
- The `Custom Setup` area of the Form can be used to override the default copyright message that will be displayed on the page.

#### Setting up the White Paper Page

When a contributor adds a document to the Document Library area—such as a product catalog, or list of “fax back” documents—it will be displayed on the Web through this page style.

- Leave the `Create this Page` field set to the default value of **Create**.
- The `Page Look` field determines the overall style and layout of the page.
- The `Background Color` button determines the page's background.
- The `Background Image` field determines which (if any) image will be used as the page's Background.
- The `Custom Setup` can be used to override the default copyright message.

This Form has a `Content Index` section that is used to define up to five categories that the people adding documents to the database can use to organize their submissions.

Enter the type of category in the `Label` fields, and the possible values for the category type in the `Choices` fields. Use the `Allow Other` fields to determine if contributors can categorize documents in categories other than those that you have defined in the `Choices` fields.



### Setting up the Document Library Security and Approvals

The Security and Workflow settings for this area are often configured so that documents in the area can be composed only by the Content Composers, and can be read by only registered users.

- The `Content Composers` field lists the people and groups of people who can add documents to this area of the site.
- The `Readers` field determines who can read the documents in this area.
- The `Number of approval steps` section of the Form details which (and from whom) approvals are required before new Library documents are available on the Web.

## GENERATING YOUR SITE

The last step is to run the software in the database that will:

- Examine your responses for each of the various answers;
- Create a database for each of the areas you identified in Step #1;
- Assemble each database by copying the requisite Forms, Views, and other design components from the Library database into each of the databases;
- Create a Home Page for each database that includes Links to all of the other areas.

After completing Step #2, use the `Generate Your Site` window in the SiteCreator database to start the process of assembling the site's databases.

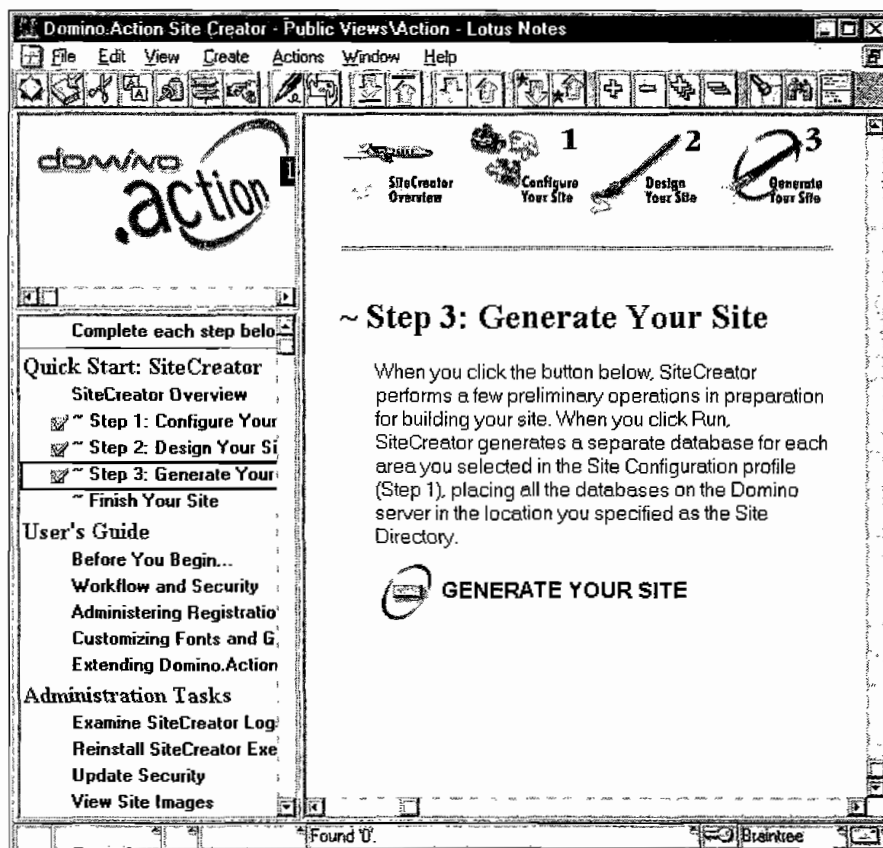


**Figure 26.22** Generating Your Site.

The SiteCreator first runs a few preliminary agents in the documents in the SiteCreator database to determine which areas you have chosen to construct and their respective Looks, Link style, and so on.

When the agents have finished, the AppAssembler window appears, which contains a button that you click to begin the actual work of assembling the site and moving the design elements of the site from the Library database into each of the site's area databases.

After clicking the **Run** button, go to lunch. The program that does the work can take more than an hour to complete, depending on how many site areas are being constructed.

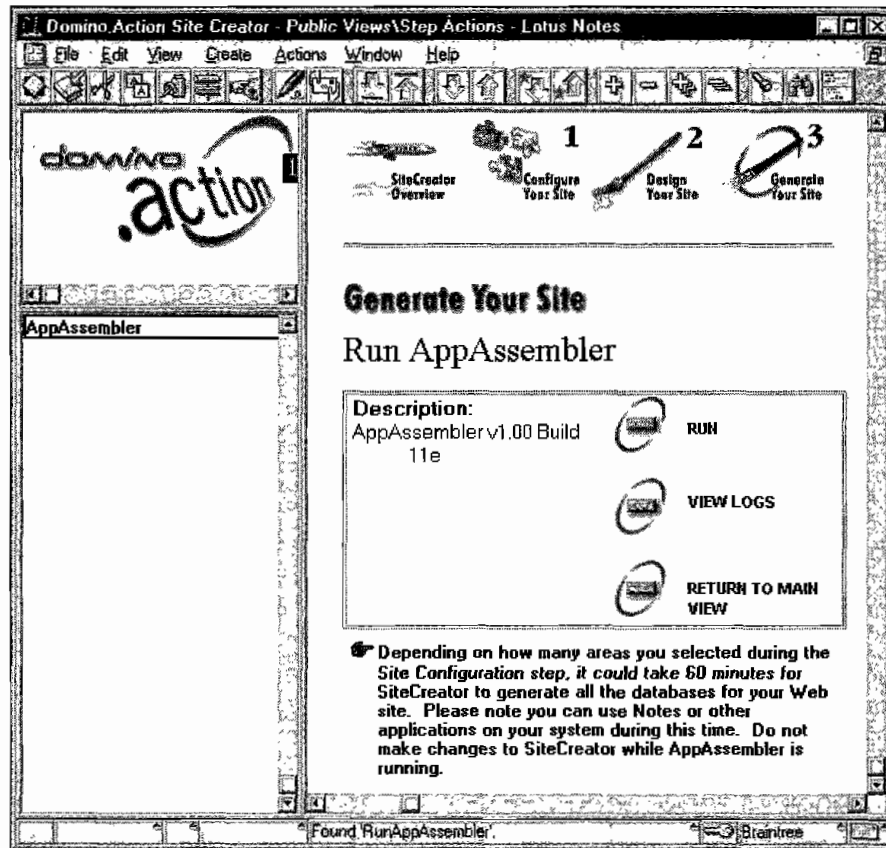


**Figure 26.23** The AppAssembler in the SiteCreator database.

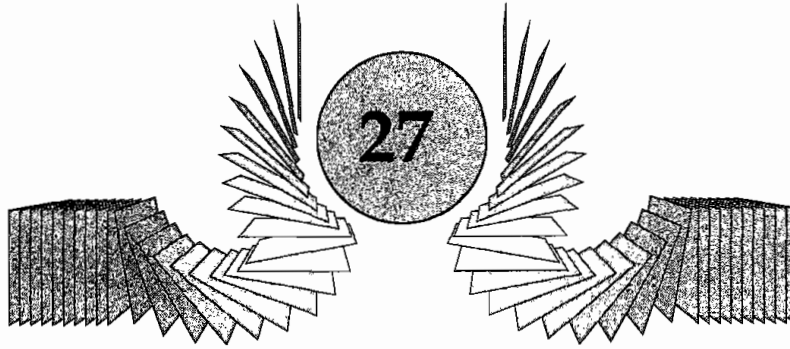
When the AppAssembler finishes, the final step in creating the site is to use the **Finish Your Site** button in the SiteCreator database. This button kicks off a final set of agents that (1) refresh the fields in all of the documents in the various area databases, and (2) ensure that the Links in all of the areas “know about” all of the other areas. This final step is especially important when you are modifying a site that has been previously built. Running these agents can take up to fifteen or twenty minutes, depending on the complexity and size of the site.

Once you have completed the **Finish Your Site** step, you are ready to interact with the site via a Web browser.

Chapter 27 and 28 detail the steps involved in securing each of the databases, establishing the content for each of the areas, and participating in the workflow processes to approve and publish the content.

## **SUMMARY**

- There are three main steps involved in producing a Domino.Action Web site: (1) Configuring the Site, (2) Designing the Site, and (3) Generating the Site.
- You should carefully plan each of the site's areas, along with the corresponding Look, Security, and Approval process.
- Using the "bulleted" or "text" link styles in the pages makes for a site with fewer graphics and faster-loading pages.



# **ADMINISTERING A DOMINO.ACTION WEB SITE**

The good news is that administering a Web site built upon a Domino.Action “foundation” is no different from the administering of any other Domino Web server. Almost all of the administrative duties and responsibilities that are required to keep a Domino.Action site running are application-related—specifically, managing the Security and Workflow settings for the site’s areas and databases.

In this chapter, you’ll get an overview of the tasks involved in administering a Domino.Action Web site, learn about refining a site’s Security and Approval Profiles, and see how to configure the site’s Approval and Notification agents.

## **THE BIG PICTURE**

As we have seen, the “constituency” for the typical Domino.Action Web site can be divided into five broad groups:

1. **The Webmaster**, who chooses the site’s look, which areas the site contains, and sets the security settings for each of the areas;

2. **The Content Providers**, who create and edit content documents—such as a Policy document, Product Description documents, and so on;
3. **The Users**, the people who are allowed to read the documents in the site that have been composed by the Content Providers, and who may also have access to create their own documents in the Discussion, Feedback, and Registration areas;
4. **Unauthenticated Users**, who typically have Author access to the Home Page and Registration areas only;
5. **The Rest of the World**, AKA the “great unwashed masses,” who may or may not have access to the site’s Home Page.

People who have been assigned the Content Provider responsibility—including the Webmaster—have access to a special Content Provider Area that is built into each part of a Domino.Action site. When such a person opens the Main page for any Domino.Action area, a special pair of Links, as displayed in Figure 27.1, appear at the top of the page.

As the figure shows, the Links are used to edit or approve the area’s Main page itself, and also to go to the aforementioned Content Provider area of the database.

Depending upon the Security Profile for a particular area in the site, users who have been assigned the Content Provider role have to use the **Log In** Link that Domino displays on the Main page of each area.

Imagine the Home Page area for the typical Domino.Action Web site. Since this area is the *entree* to the site, its security settings are usually rather lax; in fact, the Security Profile for this area may be the only one in the whole site that allows anonymous, un-authenticated users access. The upshot of such a Security Profile is that any Web user, anywhere in the world, can view the site’s Home Page. Although any user has access to the Home Page, it stands to reason that these people do *not* have access to the Home Page area’s Content Provider menu.

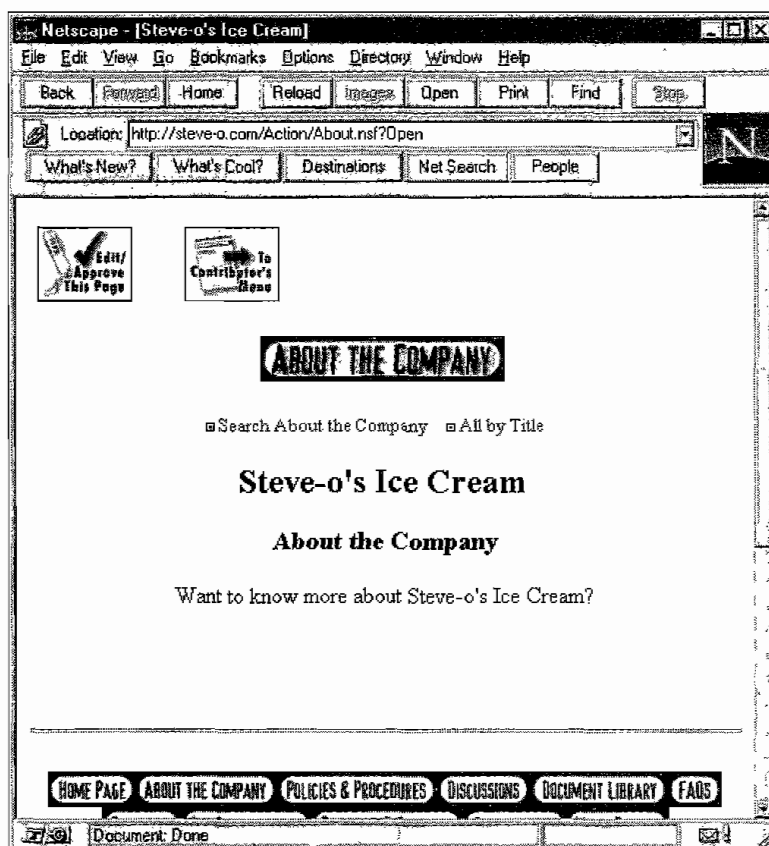


**Figure 27.1** Domino.Action providers have access to special parts of a Domino.Action Web site.

Now, meet Bill, a member of the company's Marketing Department, who has been assigned the Content Provider role, and who has the right to access the Home Page area's Content Provider menus and Views. When he retrieves the site's Home Page, he will not, by default, see the Links to the Content Provider's menu. Since the database's Security Profile grants access to any user and every user, Domino does not authenticate people requesting the page. In other words, when Bill first opens the Home Page, Domino.Action doesn't bother to find out who it is, and consequently does not display the Content Provider Links that Bill has every right to see.

The answer to this minor conundrum is for Bill to use the **Log In** Link that is automatically shown on the Main page of every Domino.Action area, and that is shown in Figure 27.2. The Links to the Content Provider areas will not appear on the page until he has done so.

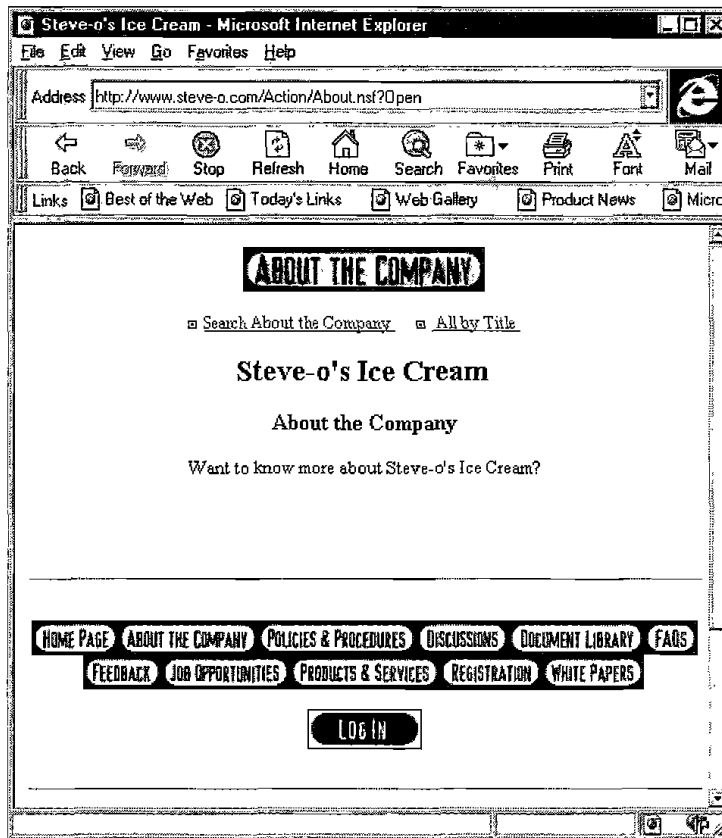


**Figure 27.2** Content Providers use the Log In Link to gain access to the restricted areas of the site.

Clicking the **Log In** Link will make Domino prompt the user for his or her Name and Password. In the previous example concerning Bill, once he enters his User Name and Password—authenticates himself to the server, in other words—Domino will recognize that he is a Content Provider, and will display the appropriate Links and grant him the proper access. This Log In Link is only an issue for the public areas in a site, for which the Access Control is set to allow the Default or Anonymous users access. In other words, if an area on the site allows



access only to the Registered users of the site, *everyone* has to log in, and Bill will never be faced with the site “not knowing it was him.”

## AREA ACCESS CONTROL LISTS

All of the security settings for each area on a Domino.Action Web site are controlled by the Security Profile for the area as defined in the SiteCreator database. In general, you should not interact with the database Access Control Lists directly; rather, use the SiteCreator to make changes to an area's Security Profile, and then update the site's Security to put the new Profile into effect.

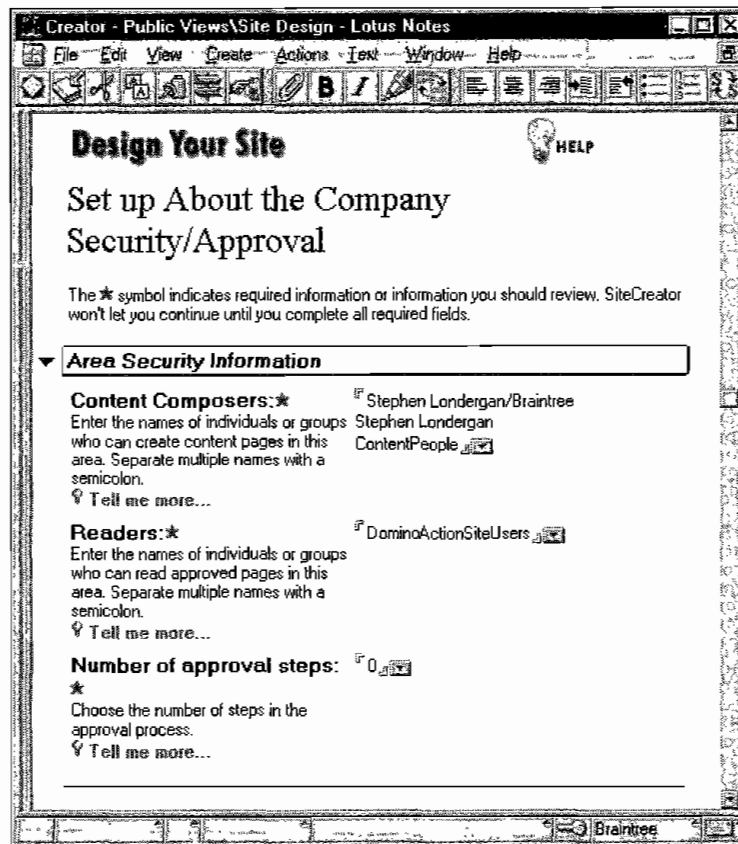


**Figure 27.3** A sample Security Profile.

Most application require at least two groups; one that is used for the area's Content Providers (listed as ContentPeople in Figure 27.3), and one that is used for the people who will have Reader access to an area.

**NOTE**

Although any group names can be used for the area's Security Profiles, the Registration area database will, by default, add newly-registered users to the group named DominoActionSiteUsers. Most administrators use this as the group for the Readers fields in the Security Profile forms.

## Updating Security

Whenever you have modified the Security Profile for an area, there are two steps to putting the new Profile into effect. Changes will *not* take effect until this process has been completed.

### Step #1: Updating the Access Control Lists

Use the **Update Security** button in the SiteCreator's Administration Tasks section to reconcile each of the area's Access Control Lists with the Security Profile documents in the SiteCreator database.

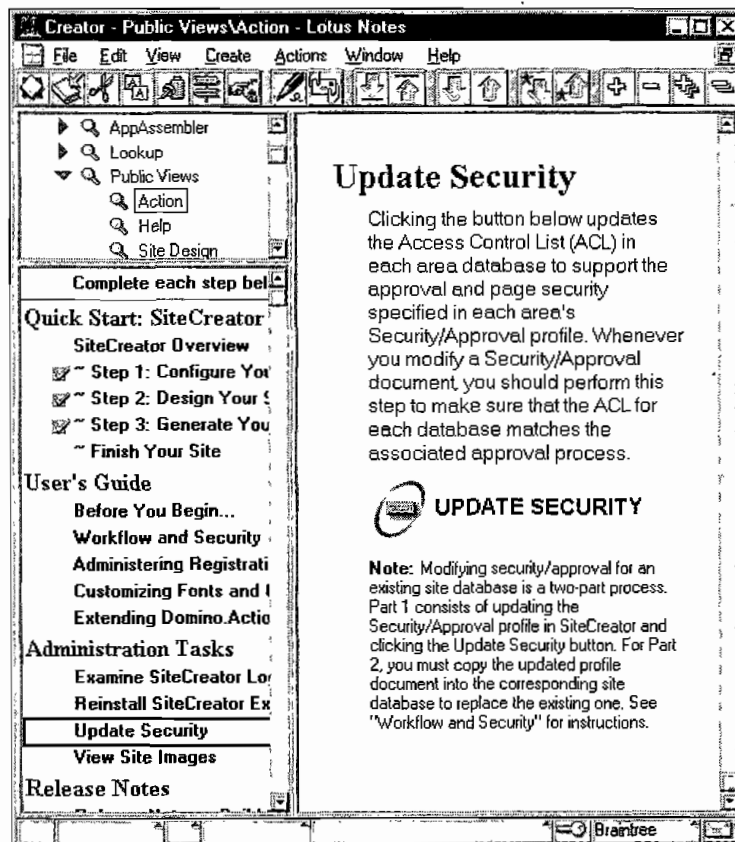


Figure 27.4 Updating a site's Access Control lists.

### Step #2: Updating the Security Profiles

The next step involved in replacing the Security Profile document in each of the area database's with the new and updated Security Profile document is in the SiteCreator database. As this process is somewhat convoluted, it's worth a blow-by-blow description of how to proceed.

First, choose **View+Go to+Workflow Profile by Area** to open the SiteCreator's Security Profiles.

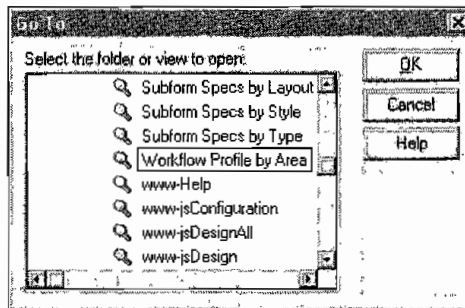


Figure 27.5 Open the SiteCreator's Security Profiles.

Select the first Security Profile document (**AboutCompany**, in the example shown in Figure 27.6) and choose **Edit+Copy** to copy it to the clipboard.

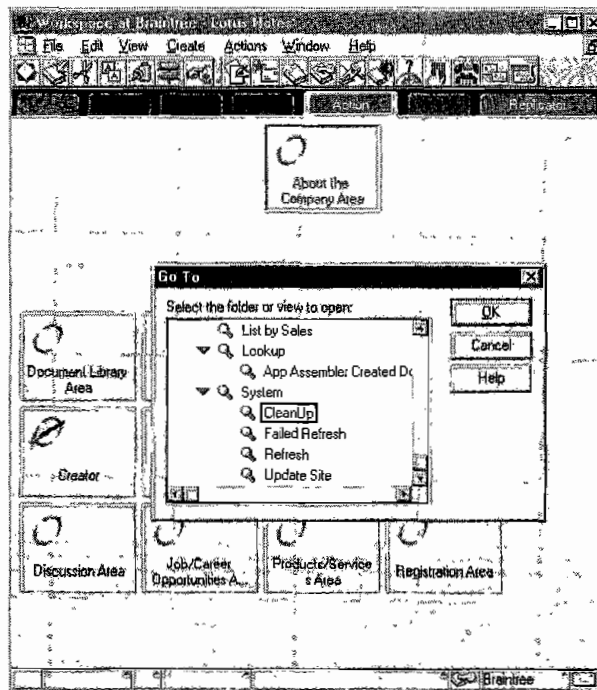


Figure 27.6 Opening an area's Cleanup view.

Select the document named **WorkflowProfile** (it's usually the last one in the view), and choose **Edit+Clear** to delete it from the database.

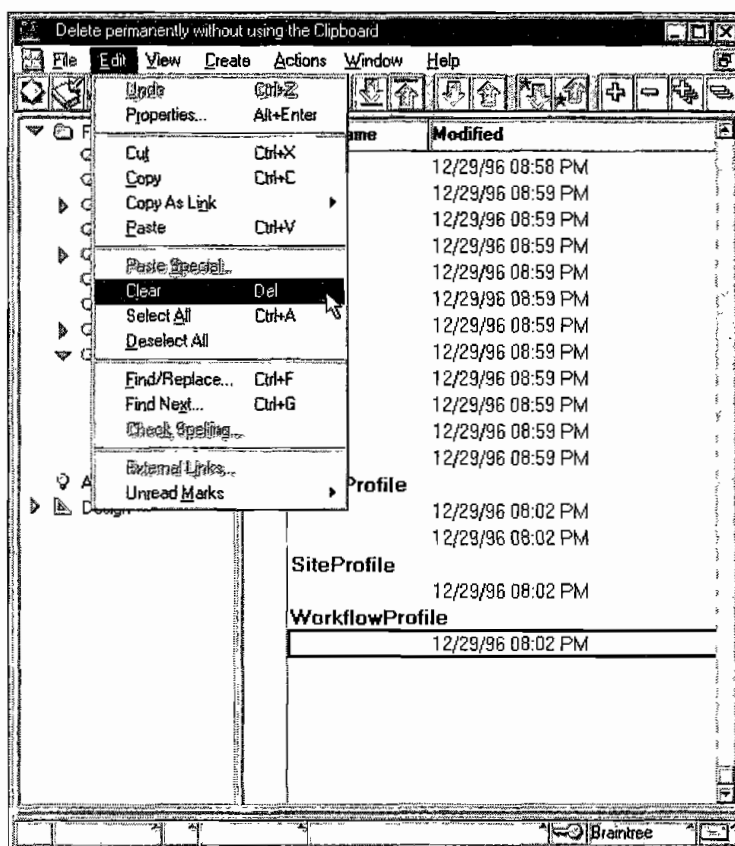


**Figure 27.7** Deleting the old Security Profile.

Finally, choose **Edit+Paste** to paste the area's new Security Profile from the clipboard.

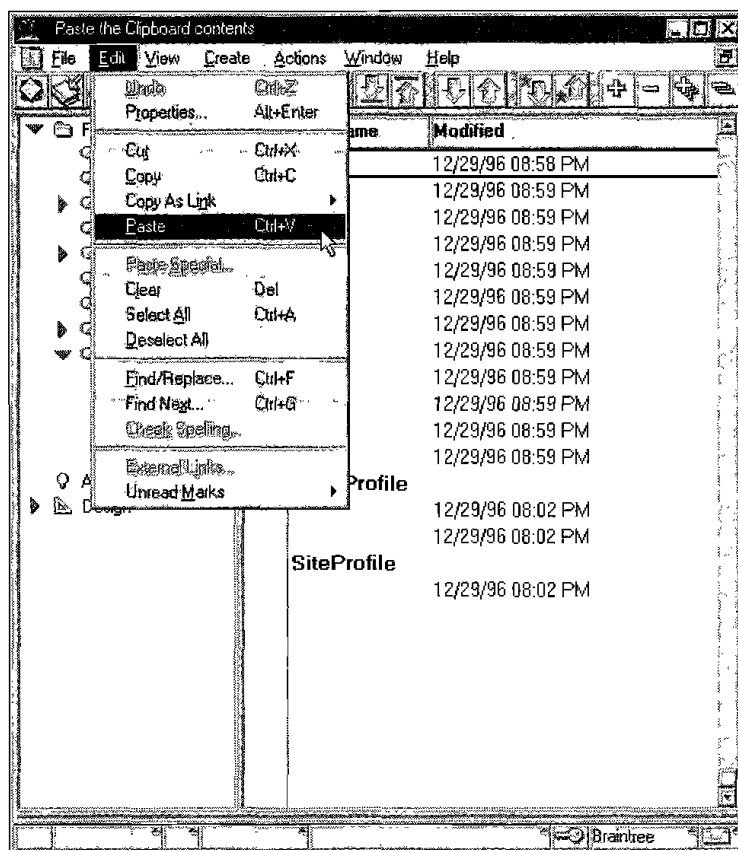


**Figure 27.8** Pasting the new Security Profile.

This process of copying, deleting, and pasting the Security Profile document from the SiteCreator database to the area database must be repeated for each database, each time the Security Profile is modified. In other words, if the site has six areas, you have to do this six times. It is admittedly an awkward undertaking, so the fewer the changes to a site's Security Profile, the better!

## The Roles

In addition to granting a person or group of people particular level of access to an area, the Domino.Action Access Control Lists include a set of Roles, detailed in Table 27.1. Although careful planning is required to determine which of these Roles is assigned to which group of users, these roles are not usually assigned manually; rather, the SiteCreator determines who gets what based on the setting in each of the area's Security Profiles.

**Table 27.1** The Domino.Action Access Control List Roles

|                   |                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Webmaster]       | This role is assigned to the person named as the <i>Site Manager</i> in the Site Configuration profile. It allows access to all areas and rights within the database/Web area.      |
| [Composers]       | Assigned to the people listed in the Security/Approval Profile's Content Composers field; allows the user(s) to create and edit new Content documents for the database/Web area.    |
| [Approvers]       | The people listed in the Security/Approval Profile's Approver fields are assigned to this role, which allows them to participate in the Approval process for the database/Web area. |
| [Readers]         | Allows a user to read pages, but not to create or edit content in the area People listed in the Security/Approval Profile's Readers field are assigned to this role.                |
| [SiteCreatorRole] | Administrative role that tracks security settings; do not delete the role or add/remove names in it.                                                                                |
| [NetPresConfig]   | Expands access to Domino.Action's design elements, used only by the Webmaster who will be customizing the SiteCreator and Library databases.                                        |

## Restricting Edit Access for an Area

The `Allow Anyone to Author` and `Allow Anonymous` fields in each of the area's Security Profiles are used to set up the databases to allow users to author, but not edit, pages.

If both of these fields are set to **Yes**, the SiteCreator will set the `Default` entry in the database's Access Control list to **Author**-level access, with the ability to create documents, and will include the `Default` entry in the `[Readers]` Role. The upshot of this is that all

visitors to the area will be able to create new pages; however, they will *not* be able to edit the pages they have created. (They will also be able to read the pages that other people have created.)

Setting the `Allow Anyone to Author` to **Yes** and the `Allow Anonymous` field to **No** means that the SiteCreator will add the `Anonymous` entry to the area's ACL, and assign it **No Access**. This means that all visitors to the area will be prompted to enter a `User Name` and `Password` before being granted access to the area, and that they will not be allowed in if they are not registered.

### Restricting Read Access

The Registration and Feedback areas need some special consideration; although users need the ability to create new pages in each of these areas, they are not allowed to edit the pages, nor are they allowed to read the pages that other users have created.

By default, the Access Control List for each of these areas is set so that all visitors—with or without a registered user name—can add documents to the area, but neither can read documents in the area. The exact settings for each the Registration and Feedback areas is determined by the Security Profile document in the SiteCreator database.

### Special ACL Considerations for Notes Workstation Users

Any user (including the Webmaster) who will be accessing the SiteCreator, Library, or any of the site databases from the Notes workstation software as well as from a Web browser, must be listed in each of the Access Control Lists twice: by the hierarchical name—for access in Notes—and by the common name—for access via the Web.

For example, Figure 27.9 shows the Security Profile in a sample Domino.Action database; one of the Content Providers is listed twice. Elaine Donnelly is listed by her common name so that she will be able to access the area via the Web. She is also listed with her hierarchical name (Elaine Donnelly/Steve-o) so that she will be able to access the database from her Notes workstation.

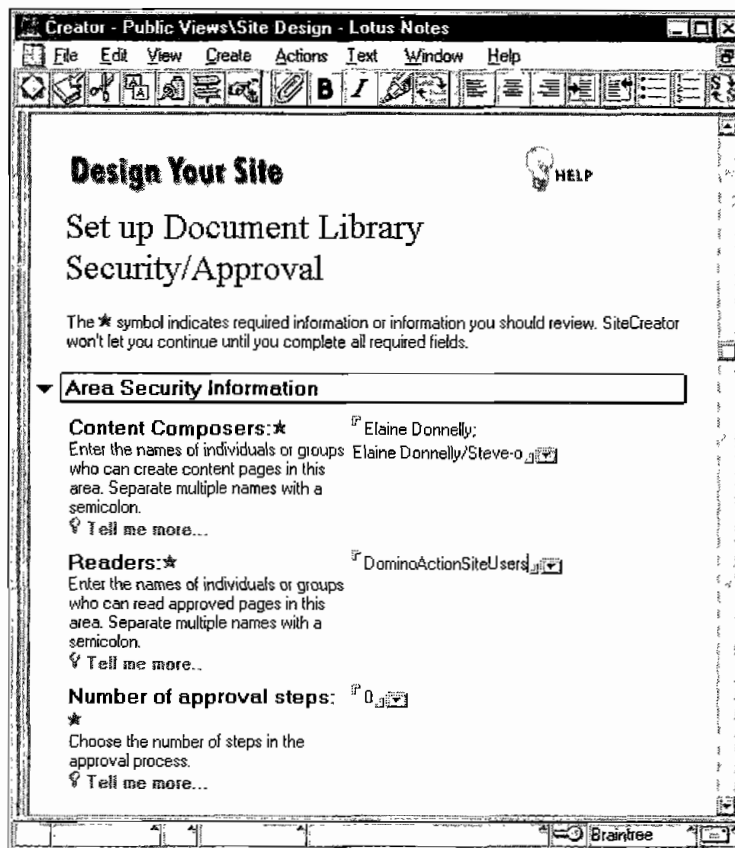


**Figure 27.9** Notes workstation users must be listed twice in each of the Area database's Access Control lists.

The two names that are required for Notes workstation users must be included as such in each of the database's Access Control Lists, and also must be listed in the `User Name` field on the user's Person document in the Public N&A Book. If access to a database is granted through group membership, the Notes User's Name must be listed both ways in the group document as well.



## PREPARING THE SITE FOR USE

There are just four steps to preparing a new Domino.Action site for use by Web users:

1. Editing each area's Main page
2. Creating a Full-Text Index for each area
3. Making the Home Page area the Home Page for the Domino server
4. Enabling the Approval agents

### Editing the Main Pages/Home Page

Each of the area's Main pages, by default, has a generic title along the lines of Click "Edit/approve this page" to enter a title. The first step to preparing a site for use on the Web is to edit each of the area's Main pages so that its Title and Subtitle are more descriptive and germane to the area.

**NOTE**

Depending upon the way a site's content will be produced, the Webmaster may not be responsible for editing and approving the Main page for each area. At the very least, however, the Webmaster has to edit and approve the Main page for the Home Page area database, as this page is the Home Page for the entire site.

To edit the site's Home Page, retrieve the page by entering a URL along the lines of *http://www.steve-o.com/action/homepage.nsf*, and then use the **Log In** Link to sign in.

**NOTE**

When the Domino.Action site was designed, one of the first questions was for the name of the site's Webmaster. This is the user name that must be used to edit the site's Home Page.

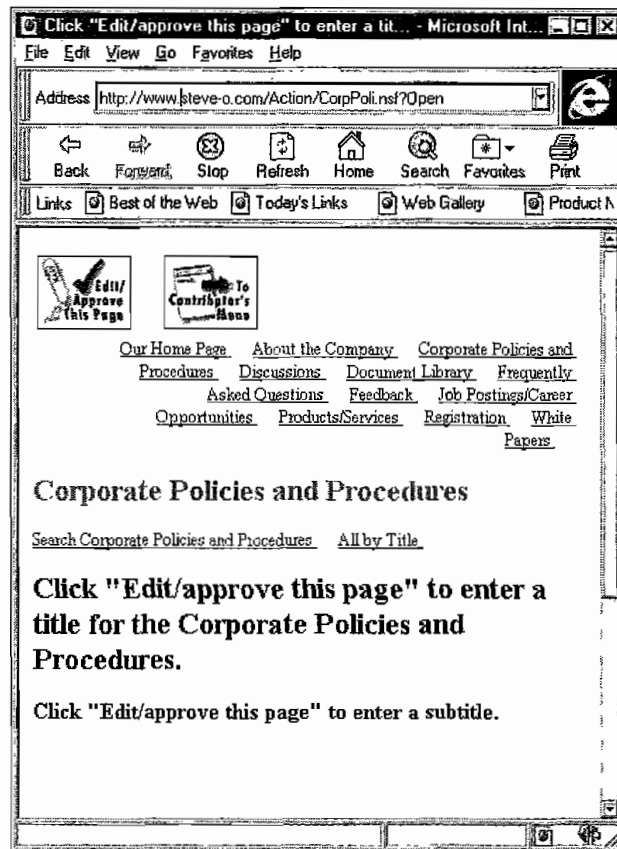


**Figure 27.10** A new Domino.Action site's Home Page.

Click the **Edit/Approve this page** Link at the top of the page to edit the page, and then replace the values in the **Title** and **Subtitle** fields on the Form. Click the **Process** Link and **Submit** button at the bottom of the page to save the changes.

This same process should be repeated for each of the Main pages for each of the site's areas and databases.

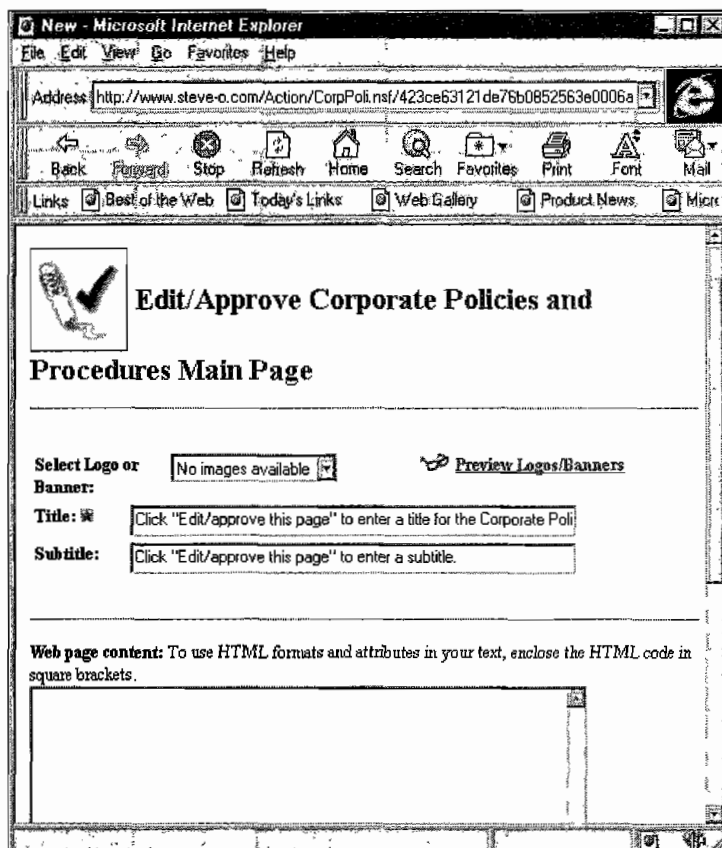


Figure 27.11 Editing a new Home page.

## Creating a Full-Text Index

Each of the areas on a Domino.Action site has built-in Links that are intended to be used to search the database. Before users will be able to perform these searches, however, the Domino.Master must create an index for each of the databases, using the Notes workstation software.

To create an index, select the icon for the area, and choose **File+Database+Properties**. Click the **Full-Text Index** tab in the Properties box, and then use the **Create Index** button to create the index. This process must also be repeated for each of the databases that make up the Domino.Action site.



TIP

You can use the **File+Tools+Server Administration** dialog boxes to quickly create multiple indexes.

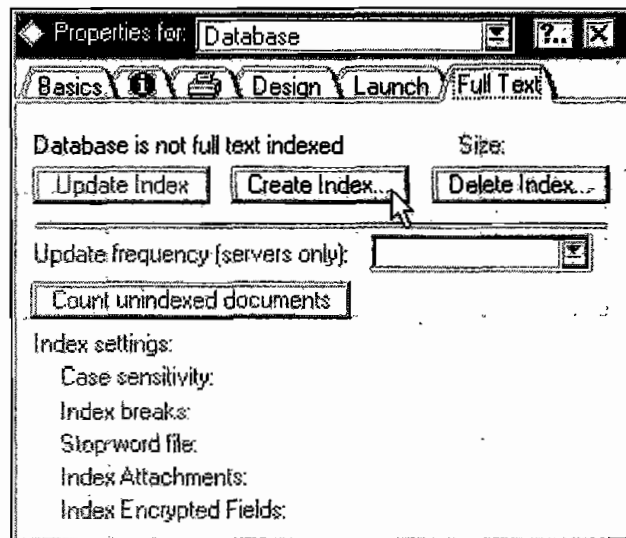


**Figure 27.12** Create a Full Text Index for each of the area's databases.

## Making the Home Page the Home Page

In most cases, the Home Page area database will serve up the Home Page for the entire site, so the Domino server must be configured to use this database as its Home Page.

Edit the Server document in the Public N&A Book to enter the name of the Domino.Action Home Page area's database in the Home URL field: typically, **action/homepage.nsf**.

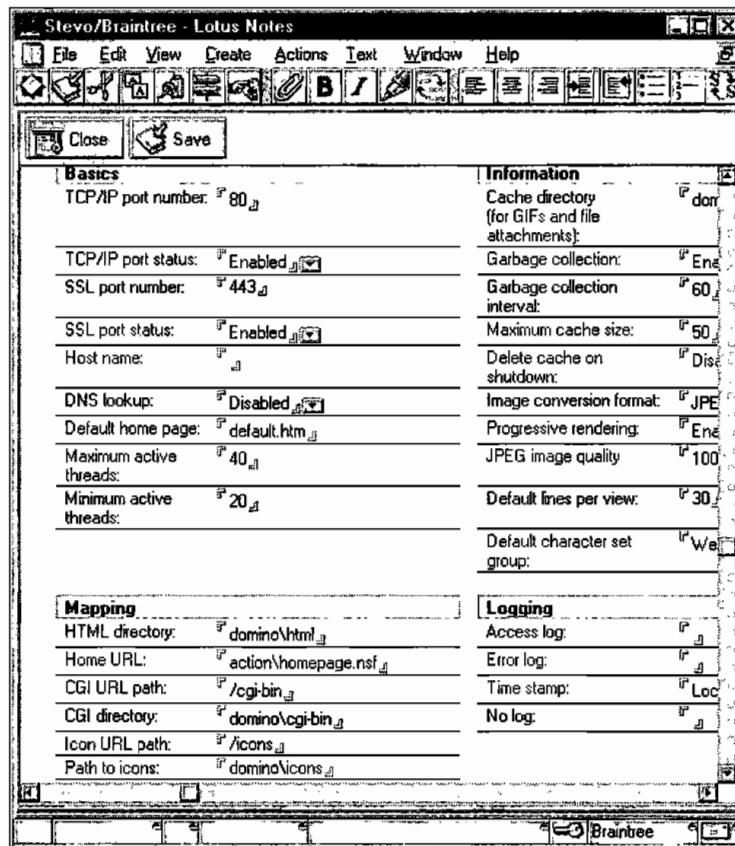


**Figure 27.13** Configuring the server to use the Domino.Action Home page area as its Home page.



**TIP**

If the Domino server is running when you edit the Server document, you must unload and then reload the HTTP task for the changes to take effect. (Type **tell http quit** and then **load http** at the Domino server console.)

## The Approval Agents

The final step in preparing a new Domino.Action Web site for use is to enable, possibly, the Approval agents that are used by the server to notify Content Providers and Approvers as new documents are added and edited on the site. If you plan to use these agents, refer to Chapter 28 for a full discussion of how they are used and enabled.

## THE REGISTRATION DATABASE

Most areas in a Domino.Action Web site are set to require a valid User Name and Password. At the very least, this means that each authorized user of the site must have a Person document in the Public N&A Book that contains his or her Name and Password, and often means that the authorized users are members of a group that is in turn referenced in the various Access Control Lists for the databases on the site. In the simplest sites, the creation and maintenance of User Names can be left as a manual task, with the Domino.Master taking on the responsibility of manually adding the required Person documents. If, on the other hand, you choose to include a Registration area in a Domino.Action site, the database comes ready-made with two agents to automate the process of adding and maintaining User Names and Passwords.

### The Handle Requests Agent

This agent processes the Registration and password-change requests that Web users post in this area of the site. When a new user requests a User Name and Password, this agent automatically creates the appropriate Person document in the Public N&A Book, and also adds the user's Name to the group named `DominoActionSiteUsers` (unless you chose a different group name in the Site Design document for the Registration page when designing the area). When an already-registered user requests a password change, this same agent will replace the old Password with the new one in the Password field in the user's Person document.

### The Send Mail Agent

The other agent in the Registration Area database is used by Domino to send an e-mail message to the user who has requested a new name or changed password. This agent produces one three e-mail messages for the user, depending on the type of request submitted and the outcome of the registration process.

If the Handle Requests agent ended normally after registering a new user, the Send Mail agent produces a message that includes the text `Thank you`

for registering with *http://yoursite.com*. Your username is username. Please remember that your password is case-sensitive.

If the Handle Requests agent successfully processed a user's request to change a Password, the Send Mail agent produces a message that includes the text Your password has successfully been changed. Your username is username. You may now log in with your new password. Please remember that your password is case-sensitive.

If the Handle Requests agent was unable to process the new user name or Password-change request for some reason, the Send Mail agent produces a message that includes the text There was a problem with your registration request. The error returned was: error. Try resubmitting your request. If you get another error notification, contact the Webmaster. Include the text of this message and describe the problems you are experiencing.

The most common reason that the Handle Request agent is not able to process a new User Name request is that the name being requested is already in use; the Handle Request agent will also fail if a user has requested a new Password, but did not supply the correct old Password.



Note that mail routing between the Domino server that will run the Send Mail agent and the Internet must be enabled for the aforementioned notification messages to be sent; the Execution Control List must also allow the Domino server to run the agents, and the ACL for the server's Public N&A Book must grant the Domino server at least Editor access.

## Running and Modifying the Agents

The Handle Requests and Send Mail agents are built into the Registration area database, but they will not run until they are each enabled with the Notes workstation software.

To configure the database so that the agents run regularly, select the icon for the Registration area and choose **View+Agents**. Click the little box immediately to the left of the Agent's name to turn it on, and then use the Choose Server To Run On dialog box to indicate the name of the Domino server that will process requests and send the e-mail messages.

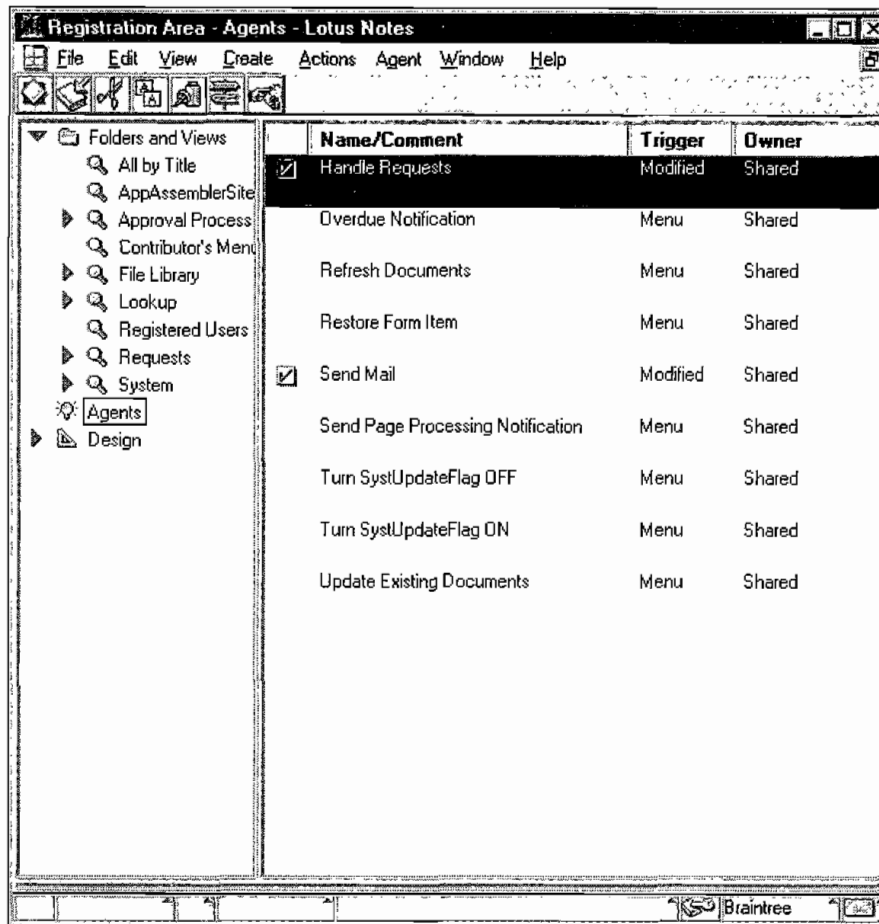


**Figure 27.14** Enabling the Process Requests and Send Mail agents in the Registration Area database.

Each of these agents can be configured to capture information about its execution to the regular Notes Agent Log file. To turn this feature on, edit the each of the agents and select the (Declarations) event. Enter the file name of the Agent Log database (in quotes) after the select the `Const LogFile=` statement. You can also use the `Const VerboseLogging=` statement to determine whether the log messages that are stored in the database named in the `Const LogFile=` statement are Verbose or Terse. Set the `Const VerboseLogging=` statement to **True** to store the most information each time the Handle Request agent runs; set it to **False** to store a relatively terse, one-line description of the agent's activities.



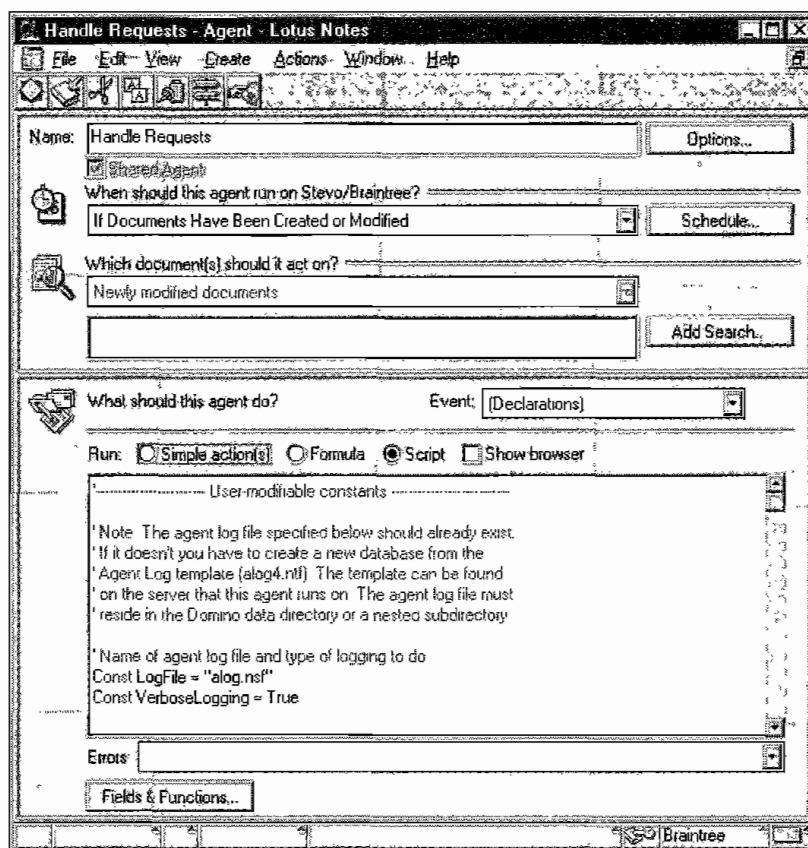


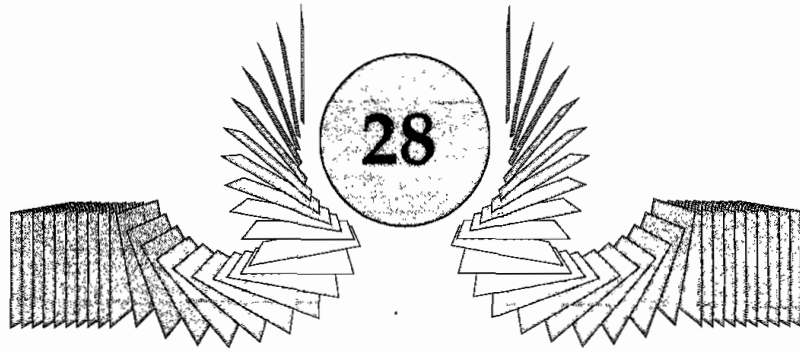
**Figure 27.15** Defining the file name and logging style for the Registration agents.

The Send Mail agent can be further modified to include the name of a person which will be included in the message that is sent when the Handle Requests agent is unable to process a user's request. Enter the name of the person (**webmaster@yourplace.com**, for example) inside the quotes after the Send Mail agent's `Const helpEmail=` statement.

Finally, the messages that are produced as the agents run can be directed to the Domino server console, instead of to the Agent Log database. Change the `AgentDebug` variable in the `Initialize` event for each agent to **True** to turn on this feature.

## SUMMARY

- Each area's Security and Approval settings are determined by the Security/Approval Profile document in the SiteCreator database.
- You must enable the Approval, Handle Requests, and Send Mail agents.
- Every Domino.Action area database should have a Full-Text Index.



# **USING AND CUSTOMIZING A DOMINO.ACTION WEB SITE**

A Web site based on Domino.Action is easy to manage, and even easier to use. The good news is that the databases that make up the site's areas come ready-made with the agents and processes required to produce the site's content.

In this chapter, you'll learn more about the Domino.Action Workflow model, how to customize the database's appearance, and how to use the Contributor's menu to upload and manage files.

## **PRODUCING WEB CONTENT: THE APPROVAL PROCESS AND PROFILE**

When a Domino Site is created with the SiteCreator, each area has the option of having an Approval Process defined for the pages that will be posted by the Content Providers to the area. It may be an organizational policy that all pages destined for the Web must first be approved by the Marketing department, or the Domino.Action Webmaster may institute a policy in which he (or she) reviews all pages before they are published. (In the case of a site that does not require pages to be

approved before publication, enter 0 in the Number of approval steps field. When this field is set to zero, a page will be available in the Domino.Action area on the Web as soon as it is saved by its author.)

The screenshot shows the SiteCreator application window titled 'Creator - Public Views\Site Design - Lotus Notes'. The main content area is titled 'Set up About the Company Security/Approval'. It includes a note that stars indicate required information. The 'Area Security Information' section contains fields for 'Content Composers' (Stephen Londergan/Braintree, Stephen Londergan, Elaine Donnelly), 'Readers' (All), 'Number of approval steps' (1), and 'Automatic notification' (Yes selected). The 'Approver Names and Function' section contains a table with one entry: Step 1, Approver: Stephen Londergan/Braintree, Function: the big man.

| Approver                                              | Approver Function:    |
|-------------------------------------------------------|-----------------------|
| Step 1<br>★<br>Approver: Stephen Londergan /Braintree | Function: the big man |

Figure 28.1 A sample Approval Process document in the SiteCreator database.

## Defining an Approval Process

To direct that pages must be approved before being published to the Web, indicate the number of Approvers by entering the appropriate number (the maximum number of Approvers for a page is five) in the Number of approval steps field.

Use the Automatic Notification field to determine whether or not Domino should automatically send e-mail notification to page authors and Approvers as a page winds its way through the Approval Process.

When a new document has been posted to a Domino.Action area that has an Approval Process, Domino sends an e-mail notification to the next person who is named in the Approvers section; it is this person's responsibility to open the page and either approve or reject it. (If the site is not using Automatic Notification, the person(s) defined as the approver(s) has to regularly check the Content Providers menu in the database area.)

The Notification messages and Interval section of the Approval Profile Form tells Domino how long a page can be "languishing" in the database before a special e-mail message is sent to the approver; the fields in the Messages section contain the actual text of the messages that will be sent by Domino, and can be modified as appropriate.

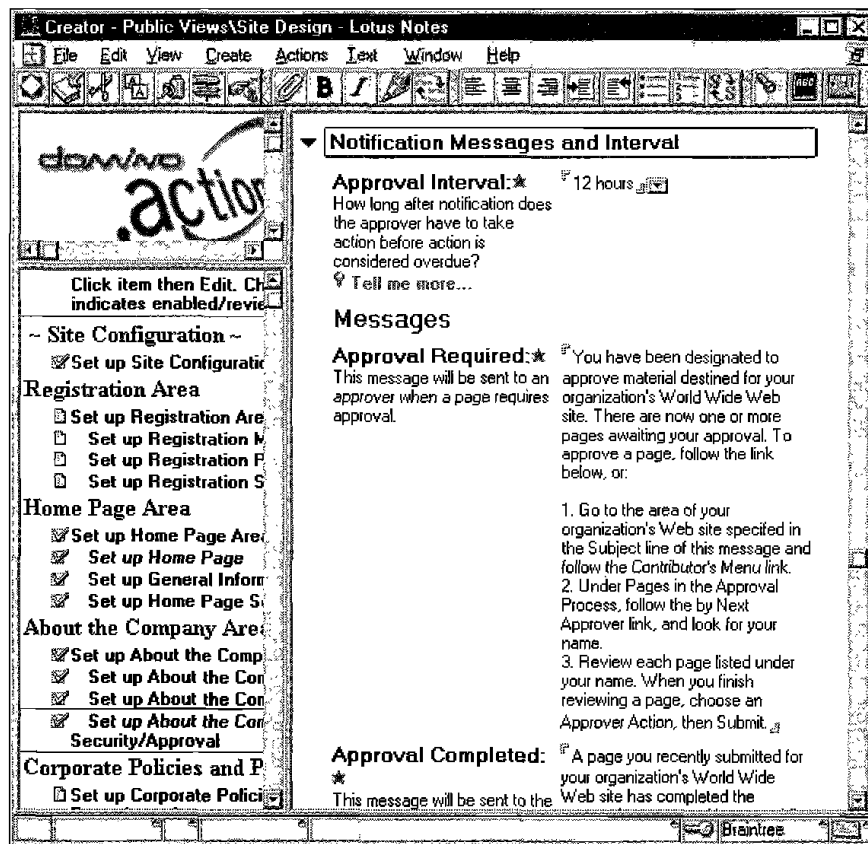


**Figure 28.2** The Messages section of the Approval Profile contains the text of the messages that are sent by Domino.

## Approval Notification

When you choose to use the Workflow/Approval Process in any of the Domino.Action area's databases, two agents are used by Domino to produce e-mail messages that keep the area's Content Providers apprised of the status of the documents they have added to the site, and also to notify an Approver if a page has not been approved within the time defined in the area's Security Profile document.

The first `Send Page Processing Notification` agent produces an e-mail message under the following conditions:

- **A page requires Approval:** When the Approval step indicates that there are several Approvers, each will receive the notification e-mail, but only one person is required to Approve the page.
- **A page has completed the approval process:** The agent will send a message to the page's originator once the page she submitted has completed the Approval Process.
- **A page has been rejected:** The agent sends a message to a page's originator if the page has been rejected.
- **A page has been rejected and returned for further editing:** The agent sends a message to a page's originator if any of the Approvers have returned it for "more work."

When the `Send Page Processing Notification` agent produces a message, it includes a Link to the document in question, as is shown in Figure 28.3.

The `Overdue Notification` agent is used solely in the case of a document that is languishing in the Approval Process longer than it is supposed to be, as indicated in the area's Security/Approval Profile.

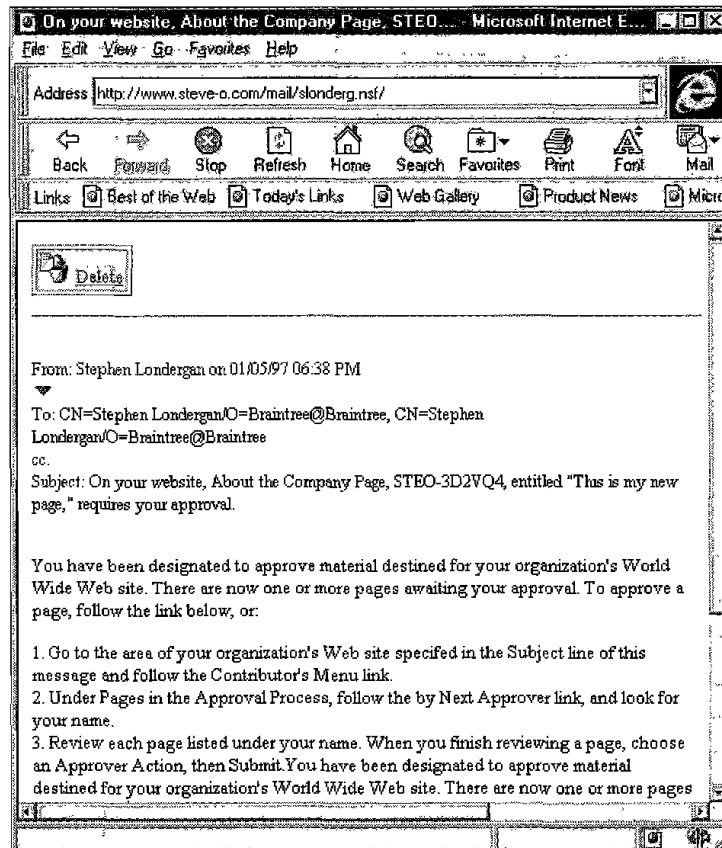


**Figure 28.3** A sample notification message, created by the Send Page Processing Notification agent.

### Enabling the Approval Notification Agents

The two aforementioned agents must be enabled and scheduled, *in each database/area*, for Approval Notification. In other words, if your site has four areas taking advantage of Approval Notification, you have to turn on the two agents four times.

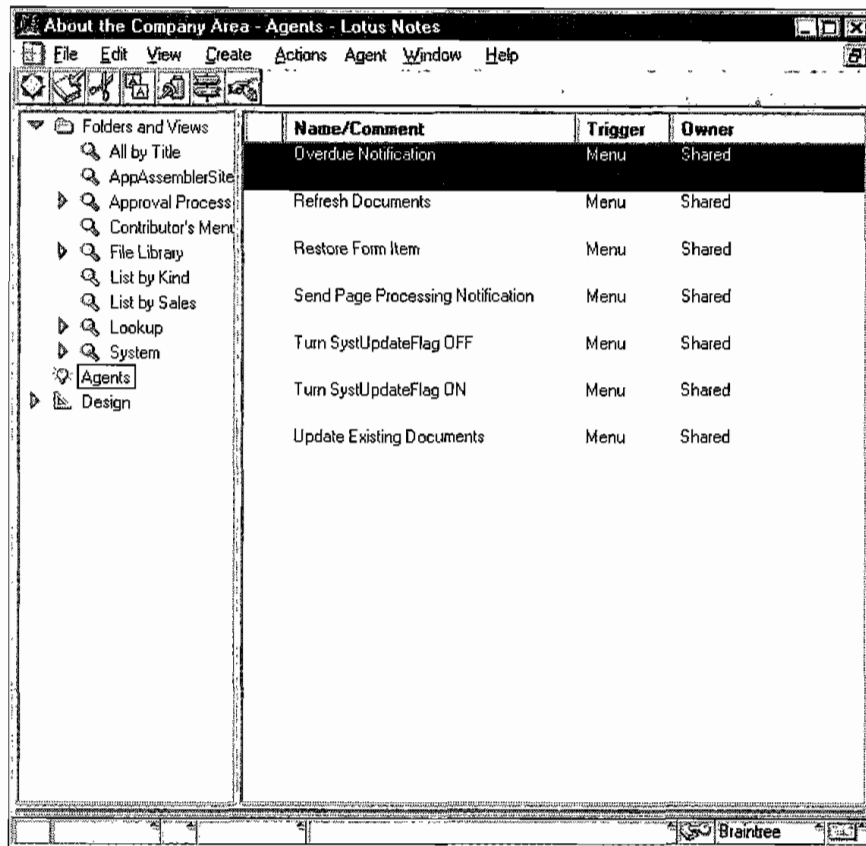


**Figure 28.4** Edit each of the Approval Processing agents to schedule the frequency with which e-mail messages will be sent by Domino.

To turn on the Approval Notification agents:

1. Select the database with the Notes workstation software.
2. Choose **View+Agents**.
3. Select either the **Overdue Notification** or **Send Page processing Notification** agent and choose **Actions+Edit**.
4. Select the frequency from the **When should this agent run** list.
5. Save the agent.

Repeat these five steps for the other Approval Notification agent, and then repeat the entire process for each database/area on the site that is using Approval Notification. Figure 28.5 shows the Overdue Notification agent being modified to run every hour.



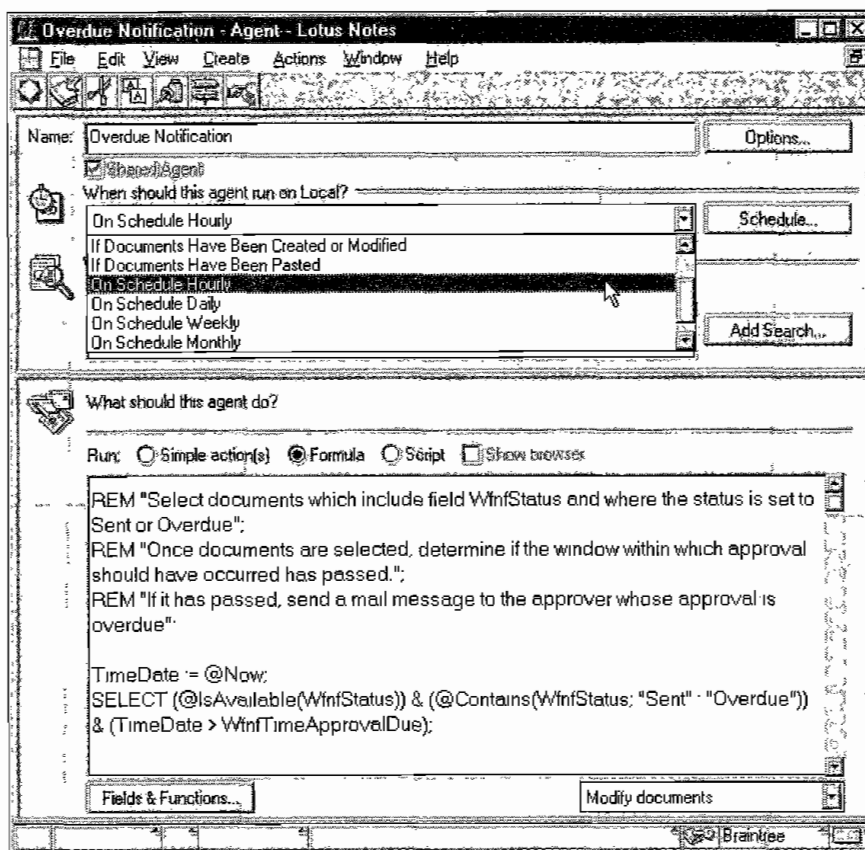


Figure 28.5 Scheduling the Overdue Notification agent.

## USING THE CONTRIBUTOR'S MENU

Any Web user who is listed as a Content Provider for an area and who has been authenticated by the Domino server has access to the Contributor's menu, which is a special part of the area that is used to:

- Edit the area's Main page
- Add new pages to the area
- Review the pages that have already been added
- Review the Approval Process
- Upload files
- View and edit the File Library

Figure 28.6 shows a sample Contributor's menu.

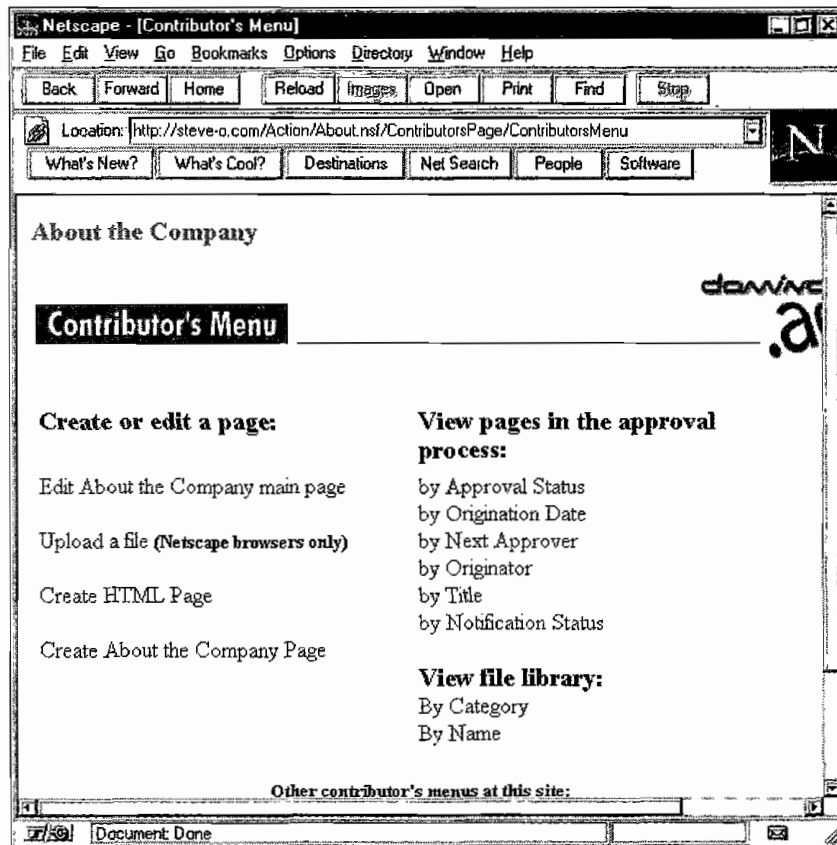


**Figure 28.6** Each area on a Domino.Action site has its own Contributor's Menu.

## Editing the Area's Main Page

As we have seen, the Main page for an area is the page that is displayed to a user who opens the area. It typically contains the title of the area, some descriptive text about the content, Links to Views for the area's content, and Links to the other areas on the site.

Content providers can use the **Edit area name Main page** on the Contributor's menu to alter the appearance of the area's Main page. Even HTML codes (enclosed in [square braces]) can be included in the page. As with any other content submission, the Content Provider uses the **Submit** button at the bottom of the page to post the document. The new Main page will either be available instantly (in the case of an area

that does not use an Approval Process), or will be saved in the Approval Process area, pending review and approval, in the case of an area that does use the Approval Process. Figure 28.7 shows a Main page being edited.

If the user clicks the **Process** button before clicking the **Submit** button, the page goes to the Web or next Approver; if she (or he) chooses to instead **Hold** the page, it is saved in the database for later editing by the same person—and is not submitted to the Web or next Approver.

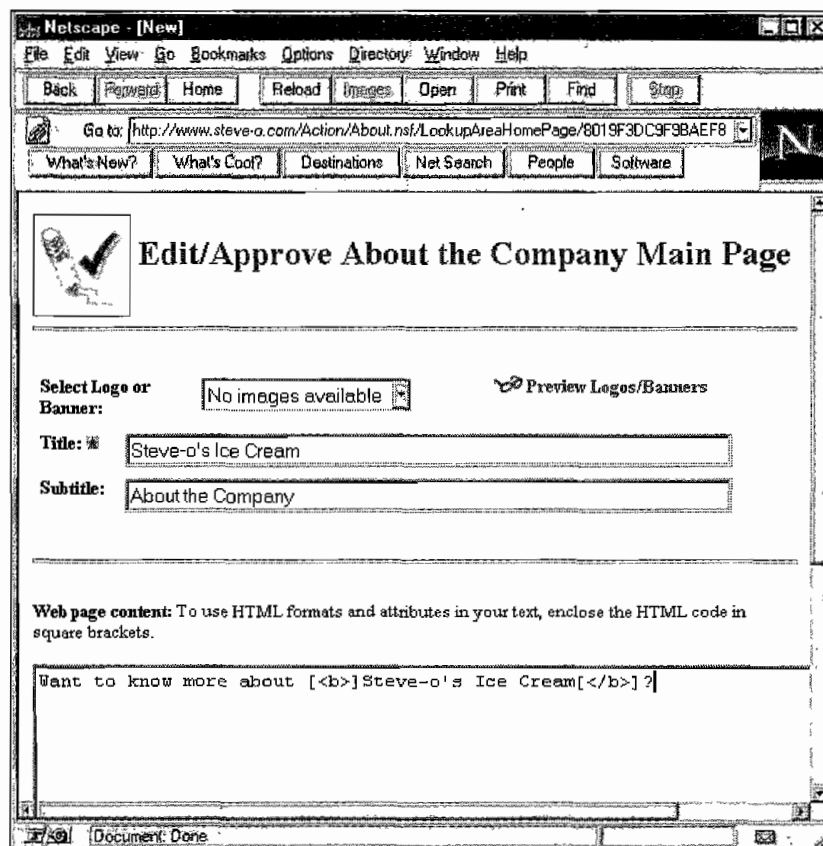


Figure 28.7 Editing an area's Main Page.

## Adding New Pages

Any one of an area's Content Providers can also use the **create HTML page** and **Create area name page** Links to add a new page to the area. Submitting such a page is similar the process of editing the area's Home

Page; after the new page is submitted, it will either be available on the Web or stored, pending approval. Depending on the area's Profile, the page may have fields that can be used to categorize or otherwise identify the new page. Figure 28.8 shows a sample "About the Company" page being submitted.

**Netscape - (New)**

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find Stop

Location: <http://steve-o.com/Action/About.nsf/3e44517f4e0cab1e85256416007fa0ba?OpenForm>

What's New? What's Cool? Destinations Net Search People Software

**Title:** We have moved our pastures

**Subtitle:** (but we're still in Vermont!)

This section lets you assign your page to one or more categories, which will help organize the information for easy access. Before you can submit this page, you must select a value (or enter a new value, if that is an option) for at least one category.

**Kind:** Press Release Enter additional Kind: FYI Document

**Sales:** New Products

**Web page content:** Text or [HTML]

**Web page content:** To use HTML formats and attributes in your text, enclose the HTML code in square brackets.

Since 1995, we have fed and grazed our herds in Alberta. But steadily increasing feed prices and the unhospitable (and very ro forced us to change the plan--so now he cows have a new home!

Document Done

**Figure 28.8** Adding a new page via the Content Provider's menu.

## Working with the Process Views

Users listed as Content Providers in an area's Security/Approval Process can also use the Views to review the pages that have been submitted, and, if appropriate, approve and/or edit them. Figure 28.9 shows the by Originator View from a sample "About the Company" area.

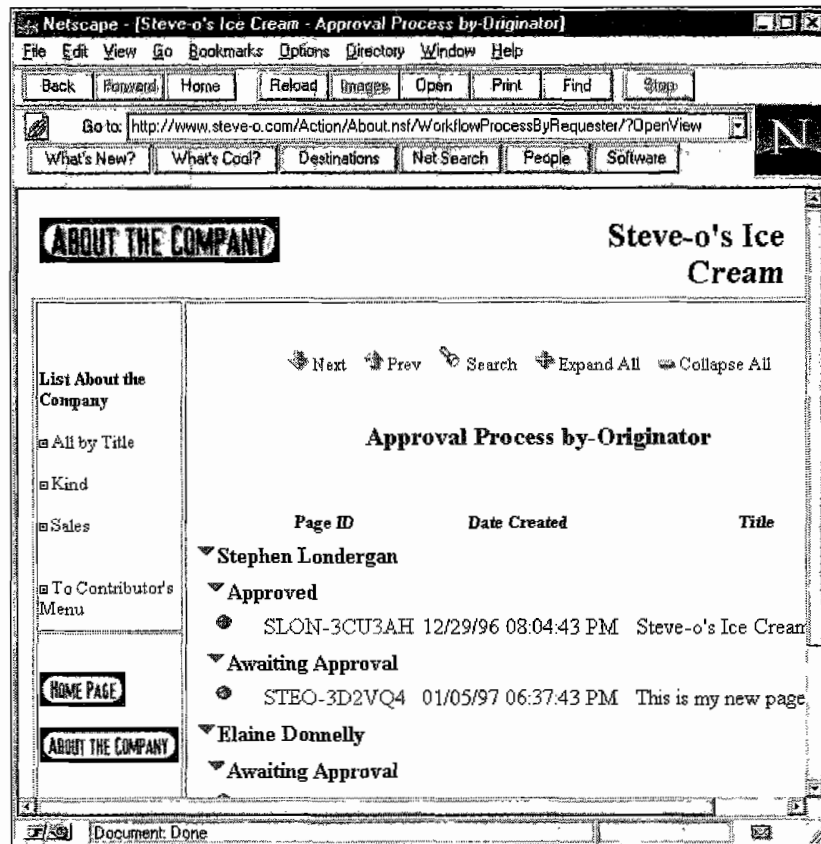


**Figure 28.9** The Content Menu has Links to several views that show the content pages in the area.

## Approving & Rejecting Pages

Depending on the individual user's status as a Content Provider, she (or he) may be called upon to approve pages that have been submitted by others. When the user who has been defined as a page's Approver opens the page from any of the Process Views, he (or she) can use the **Edit/Approve this page** button at the top of the page. Doing so allows the approving user to edit the page itself, and also use the Approval section at the bottom of the page to determine where the page is headed next—either published to the Web, sent to the next person in the approval "chain of command," or returned to the originator for more work. Figure 28.10 shows such a sample page being approved.

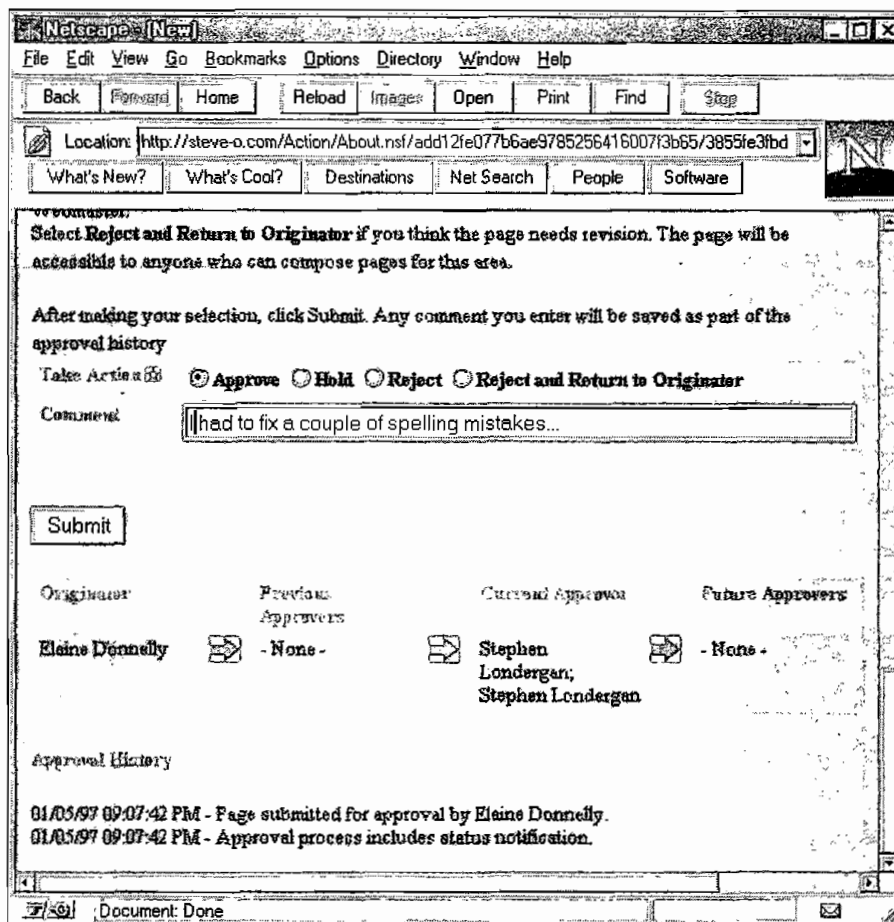


Figure 28.10 Approving a page.

## Working with the File Library and Uploading Files

All of the images that are displayed in a site—such as the banners, Link icons, and perhaps some custom images that were included in the area when the site was created—are available for viewing and/or editing via the **View File Library** Links on the Contributor' menu. These views can be used to find a particular image, or to find the URL that can be used to include the image in a document.

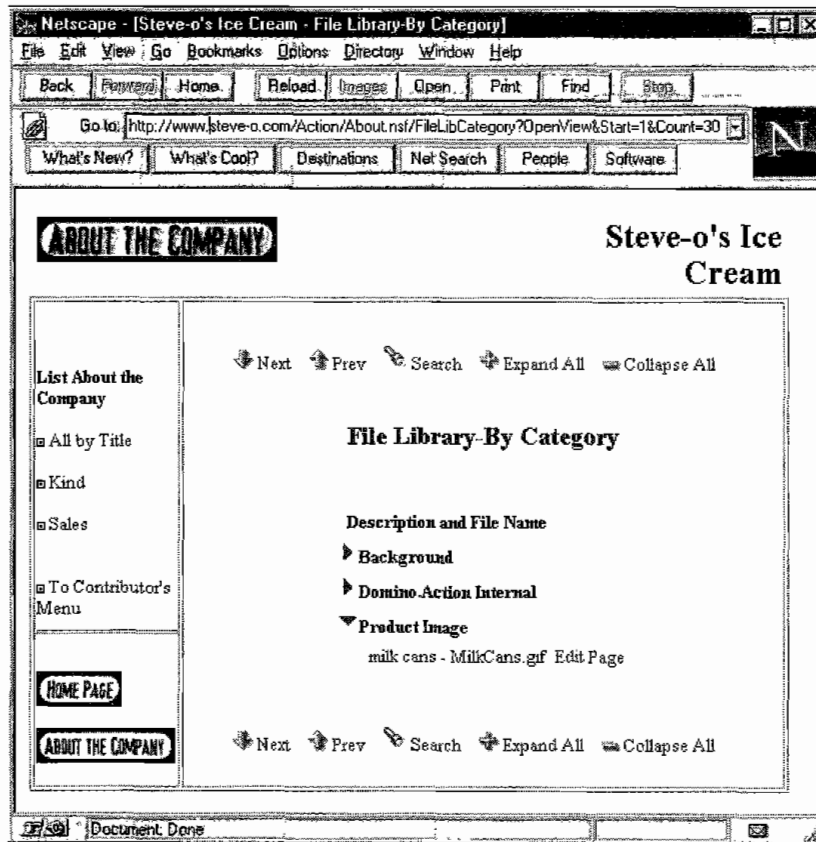


Figure 28.11 A page from the File Library.

For example, any image that was included in the site when it was created could be displayed in an area's Main document, or in one of the Content documents. Files can also be included in a site for download by users; an ISP might include custom software right in a Domino.Action area that would be available for users to download via the Web.

Use **HTML Passthrough** to include a file in a Content document. For example, Figure 28.12 shows a sample Feedback page being created, with an HTML Passthrough reference to an image file that was included in the area.

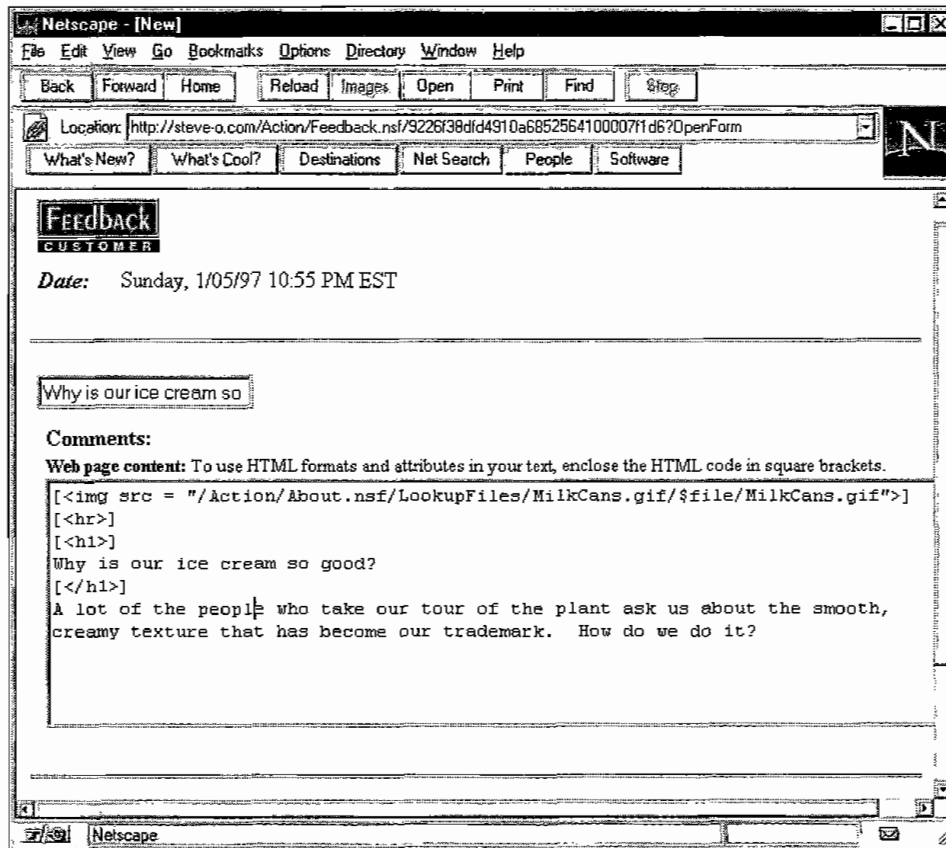


**Figure 28.12** Use HTML Passthrough to include a file in a page.



**TIP**

Always use the File Library Views to find the URL for the image that you need to include.

Content Providers who use Netscape can use the **Upload a file** Link on the Contributor's menu to upload a new file into the area, which can subsequently be included in any of the area's pages. Figure 28.13 shows a page that includes a Link to file that was uploaded in such a manner.



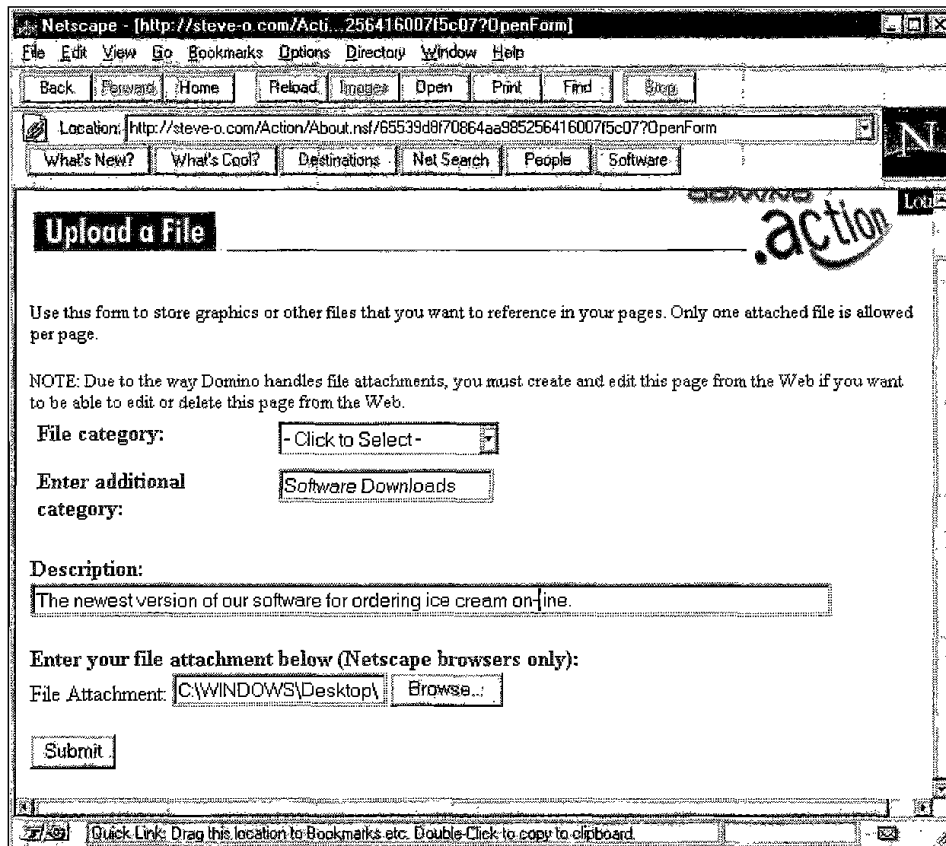


Figure 28.13 A page with a Link to an uploaded file.

When a Web user clicks the **Download it now!** Link, Domino retrieves the attached file and sends it to the user's workstation.



TIP

Use the **<A HREF>** HTML tag to create a Link to a downloadable file from the File library; use **<IMG SRC>** to display an image.

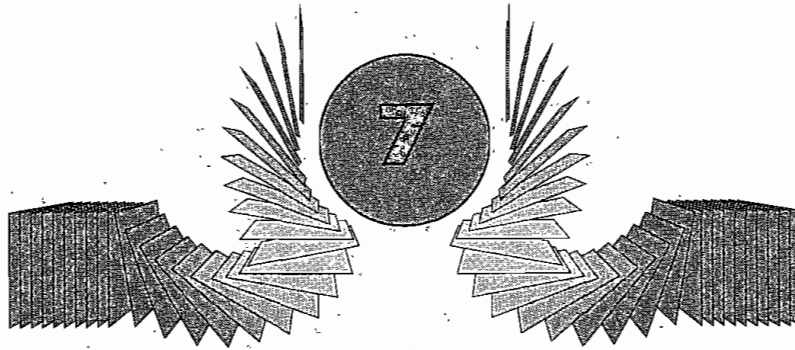
As the File Library also lists the graphic elements that are included by default by the SiteCreator, you can also use the File Library to customize an entire area's appearance, by editing the image files and replacing them with your own.

Note that all of the images are stored in all of the area databases, so if you want to change the appearance of, say, the background image that is displayed at the top of a page with the basic look, you must replace the `bbgtop1.gif` file in each of the site's areas.

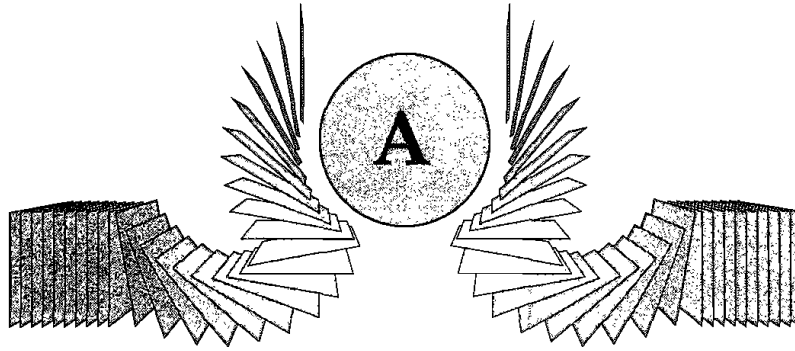
## SUMMARY

- Domino.Action Web sites have built-in Workflow processes.
- Agents in the databases automatically monitor and route documents through the Approval Process defined in the SiteCreator.
- Netscape users can upload files to a Domino.Action site via the Web.

# SECTION



# APPENDICES



# **MORE ON THE INTERNOTES NEWS GATEWAY**

## **CHOOSING A USENET SERVER**

As we discussed throughout Part 4 of this book, the decision about with which Usenet server(s) your InterNotes News Gateway will exchange articles is an important one.

The first (and most important) consideration concerns which newsgroups are available on the Usenet server. If you need to subscribe to any of the *clari.\** hierarchy, for example, you'll obviously need to make sure that the Usenet server with which your InterNotes News Gateway exchanges articles with that hierarchy. (The *clari.\** hierarchy is fee-based, so it is not available on all Usenet servers.) The bottom line is that not all Usenet servers carry all of the thousands and thousands of newsgroups that are available in the world, so make sure that the one you use has what you want.

The second consideration is the type of connection service you need; in other words, do you want your News Gateway to “pull” or to “push” articles? Although shifting the onus for news connections to the Usenet server is certainly appealing from a News Gateway performance perspective, it is not a foregone conclusion that all Usenet servers will support or allow this type of exchange. In particular, Usenet servers run by ISPs typically do *not* offer this service. If your News Gateway requires “push” service, you need to make that clear to the Usenet administrator, as there is some special configuration needed on the Usenet server. Also note that you may have to pay more for a “push” relationship with an ISP.

The third consideration concerns who owns the Usenet server. In general, you’ll find three kinds of Usenet servers: (1) those already owned and maintained by your company or organization, (2) those owned and maintained by universities, and (3) those owned and operated by commercial Internet Service Providers. Table A.1 lists a few general “pros” and “cons” for each.

**Table A.1** Ownership of the Usenet server—Pros and Cons.

| Owner/Location | Pro                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Con                                                                                                                                                                                                | Note                                                                                  |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Your Company   | <p><b>Usually free:</b> You can connect your InterNotes News Gateway server to a Usenet server at your company without incurring connect charges</p> <p><b>Easier administration:</b> Since the Usenet server’s administrator is a fellow employee (and may even be you!), configuration changes and other Usenet server management issues are (presumably) easy to resolve</p> <p><b>Push exchanges:</b> It is easiest to use configure your News Gateway for “push” article exchanges if the Usenet server is “in house”</p> <p><b>Fastest Performance:</b> Since you can usually connect the InterNotes News Gateway to the Usenet server directly via the LAN</p> | <p><b>Easier administration?</b> Since the Usenet server’s administrator is a fellow employee, configuration changes and other Usenet server management issues may be <i>harder</i> to resolve</p> | Some large companies employ charge-back mechanisms between departments and workgroups |

A Commercial Internet  
Service Provider

**Fees:** ISPs are in business to make money; you have to pay for access.

Most ISPs offer programs for a flat monthly fee with unlimited connect time, which often work out to be more economical than a pay-as-you-go plan

**Selection:** An ISP typically has the widest-possible selection of newsgroups

**Selection?** With this widest selection comes the potential headaches of users subscribing to *alt.sex*, etc.

Since there is no way to which newsgroup titles are downloaded from the Usenet server, the ACL of the News Configuration database is the only way to limit the users' ability to subscribe

**Technical Support:** Most ISPs offer technical support and assistance as part of their subscription packages

A University

**Free:** Some universities allow free connections to their Usenet servers, although they're becoming harder and harder to find!

**No Technical Support:** Who do you call when it stops working?

As the wise man once said, "you get what you pay for"

## SOME USENET NEWSGROUPS

A list of every available newsgroup would take up more pages than this book contains, and would also be out-of-date within about five minutes of being printed. None the less, Table A.2 presents a short list of newsgroups, intended to illustrate the depth and breadth of discussions raging across Usenet (and around the world). Remember that your ability to subscribe to any newsgroup is determined by what's available on your Usenet server, so "check your local listings."

The numbers under each newsgroup's description are the group's estimated worldwide readers, estimated number of messages posted each month, and the estimated megabytes produced by that newsgroup each month.

**Table A.2** Some representative newsgroups.**rec.toys.lego**

Discussion of Lego, Duplo (and compatible) toys.

12000,1787,3.2

**sci.archaeology**

Studying antiquities of the world.

23000,2191,6.1

**rec.humor**

Jokes and the like. May be somewhat offensive.

140000,4752,7.1

**misc.business.consulting**

The business of consulting. (Moderated)

7900,553,0.9

**alt.sport.darts**

Look what you've done to the wall!

6600,228,0.4

**comp.mail.maps**

Various maps, including UUCP maps. (Moderated)

21000,478,11.8

**comp.infosystems.www.browsers.ms-windows**

Web browsers for MS Windows.

13000,2063,2.8

**rec.games.pinball**

Discussing pinball-related issues.

20000,2042,3.7

**sci.econ**

The science of economics.

31000,1336,2.5

**comp.dcom.isdn**

The Integrated Services Digital Network.

30000,1569,2.4

**alt.comp.virus**

An unmoderated forum for discussing viruses.

4800,885,1.8

**biz.pagesat**

For discussion of the Pagesat Satellite Usenet Newsfeed.

2500,60,0.1

**comp.sys.mac.announce**

Important notices for Macintosh users. (Moderated)

63000

**alt.security**

Security issues on computer systems.

53000,1258,1.5

**rec.sport.golf**

Discussion about all aspects of golfing.

33000,5142,7.5

**alt.fan.douglas-adams**

Author of "The Meaning of Liff", &amp; other fine works.

23000,820,1.3

**soc.culture.polish**

Polish culture, Polish past, and Polish politics.

21000,1771,3.1

**alt.airline.schedules**

Current airline schedules of various carriers.

**alt.agriculture.fruit**

Fruit production.

6600,170,0.2

**sci.engr.television.advanced**

HDTV/DATV standards, equipment, practices, etc.

4000,296,0.7

**k12.ed.science**

Science curriculum in K-12 education.

4600,301,0.4

**comp.sys.ibm.pc.hardware.networking**

Network hardware &amp; equipment for the PC.

28000,603,0.6

**misc.transport.trucking**

Commercial trucking related issues.

3700,412,0.6

**biz.jobs.offered**

Position announcements.

40000,7417,6.1

**biz.marketplace.discussion**

Discussion of biz.marketplace issues.

**alt.sports.baseball.bos-redsox**

Boston Red Sox major league baseball.

4800,1288,1.7

**soc.couples.wedding**

Wedding planning.

7300,3592,5.7

**soc.culture.irish**

Ireland and Irish culture.

5300,3473,6.8

**comp.os.ms-windows.networking.tcp-ip**

Windows and TCP/IP networking.

28000,1648,1.9

**comp.os.ms-windows.announce**

Announcements relating to Windows. (Moderated)

55000,21

**comp.protocols.tcp-ip**

TCP and IP network protocols.

49000,772,0.9

## CONSOLE COMMAND QUICK REFERENCE

The list of server console commands available for controlling the News Gateway is refreshingly short. Remember that in most cases, the News Server is configured to load automatically when the Notes server is started—via the **servertasks=** line in the server's NOTES.INI.

- **load innews:** loads the News Gateway software
- **show tasks:** shows all of the tasks running on the server, and their status
- **tell innews abort:** cancels any transfer or process being performed by the News Gateway
- **tell innews newsgroups:** downloads the latest list of available newsgroups from the Usenet server
- **tell innews transfer:** initiates an immediate newsfeed
- **tell innews quit:** unloads the News Gateway program



## NOTES FOR THE UNIX ADMINISTRATOR

If you are already familiar with Usenet, you may be interested in the following list of differences between the InterNotes News Gateway and other Usenet News systems:

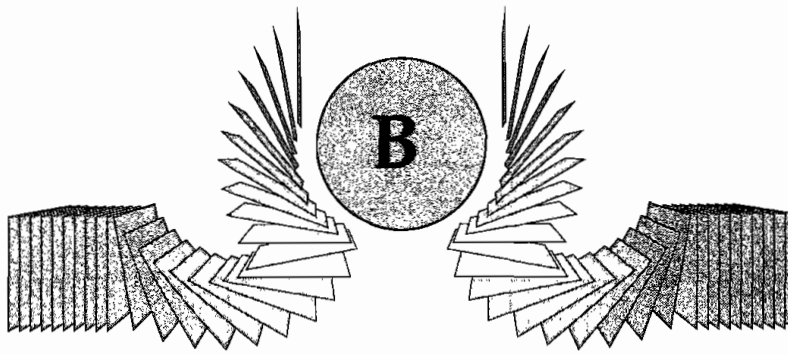
- The News Gateway has no facility for using `.killfiles` to eliminate articles from certain people or about certain topics. This functionality can be simulated by adding an Agent to the News database(s) that regularly deletes documents based on the contents of the `from` or `subject` field.
- The News Gateway has no built-in facility for encoding or decoding binary files. Incoming articles that contain encoded binary files can be exported to an ASCII file and then decoded with an external program. Similarly, outgoing binary files must be encoded with an external program, and then imported into the article when it is being created.
- Incoming articles already include fields for the article's headers and number of lines. You can modify the News database to display these fields, if required.
- The News Gateway has no facility for canceling an article once it has been moved from the Spool database to the Usenet server.

## CHOOSING AN INTERNET SERVICE PROVIDER

There are literally thousands of Internet Service providers, most of which have sprung up rather recently in response to the explosive growth of the Internet. If you choose to use a commercial service for Usenet feeds, make sure that the company you choose provides Usenet access, and specifically, access to the newsgroups you need. Most ISPs are designed primarily to provide individual users with dial-up access to the WWW, so an ISP that specializes in corporate services might be a better choice when shopping for a Usenet server for your News Gateway.

*www.lycos.com* maintains an up-to-date list of ISPs, including those that provide national service. Check them out at [http://a2z.lycos.com/Internet/Access\\_Providers/](http://a2z.lycos.com/Internet/Access_Providers/).

Alternately, your favorite WWW search engine will provide plenty of links if you search for **internet service provider**.



# **DOMINO URLs**

## **ABOUT THE DOMINO URL COMMANDS**

Domino uses special URLs to access Notes servers, databases, and the other components and design elements of a Domino Web site. In general, Web users do not often interact with a Domino site by using these URLs; rather, the Domino application designer creates formulas within an application's Hotspots and agents that generate these URLs for the user.

## **THE DOMINO URL COMMAND SYNTAX**

Domino URL commands have the following general syntax:  
*http://Host/NotesObject?Action&Arguments*

where:

*Host* = the Domino server's host name or IP address

*NotesObject* = a Notes construct (such as database, View, document, Form, Navigator, or agent)

*Action* = the desired operation on the specified Notes object (such as **?OpenDatabase**, **?OpenView**, **?OpenDocument**, **?EditDocument**, **?OpenForm**, etc.)

*Arguments* = a qualifier of the action—for example, **Count = 10** combined with the **?OpenView** action limits the number of rows displayed in a View to 10.

## Usage Notes

- In addition to opening a Domino database's design element by name, Domino supports the use of the following identifiers to identify the parts of a database: **\$defaultView**, **\$defaultForm**, **\$defaultNav**, **\$searchForm**, **\$file**, **\$icon**, **\$help**, **\$about**.
- *NotesObject* can be the database name or replica ID, or the object's name, universal ID, NoteID or special identifier. Since an object's name and universal ID are identical in all replica copies of a database, but the Notes object's NoteID is not, referring to an object by object name or universal ID is preferable.
- *Action* can be explicit or implicit: **?OpenServer**, **?OpenDatabase**, **?OpenView**, **?OpenDocument**, **?OpenForm**, **?EditDocument**, or **?Open**, **?Edit**, **?Delete**. Domino defaults to the **?Open** action if an action is not included in a URL.
- Append the **login** argument to any Domino URL to require user to enter his/her name and password.
- Substitute a plus sign (+) for spaces in a URL.
- Separate arguments with **&** (ampersands); for example, <http://www.steve-o.com/Home+Page.nsf/Products+by+Region?OpenView&ExpandView>
- When referring to a cascading View and Form name, replace the backslash with a forward slash. For example, to open a View named Parts\By Name, use a URL along the lines of <http://www.steve-o.com/products.nsf/Parts/By+Name>.

## OPENING SERVERS, DATABASES, AND VIEWS

The following commands access servers, databases, Views, About documents, Help documents, and database icons.

| Action                                                | Syntax                                                                                         | Example                                                                            |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| OpenServer                                            | http://Host                                                                                    | http://www.steve-o.com/?OpenServer                                                 |
| OpenDatabase                                          | http://Host/DatabaseFileName?<br>OpenDatabase                                                  | http://www.steve-o.com/products.nsf/<br>OpenDatabase                               |
|                                                       | http://Host/DatabaseReplicaID?<br>OpenDatabase                                                 | http://www.steve-o.com/852562F3007ABFD6?<br>OpenDatabase                           |
| OpenView<br>o.com/products.nsf/By+Name?               | http://Host/Database/ViewName?<br>OpenView                                                     | http://www.steve-<br>OpenView                                                      |
|                                                       | http://Host/Database/<br>ViewUniversalID?OpenView                                              | http://www.steve-o.com/parts.nsf/<br>EED087A8ACE170F2252542F360762264?<br>OpenView |
|                                                       | http://Host/Database/ViewNoteID?<br>OpenView                                                   | http://www.steve-o.com/leads.nsf/00000197?<br>OpenView                             |
|                                                       | http://Host/Database/\$defaultview?<br>OpenView                                                | http://www.steve-o.com/leads.nsf/\$defaultview?<br>OpenView                        |
|                                                       | Start = <i>n</i> , where <i>n</i> is the row number<br>to start with when displaying the View. | http://www.steve-o.com/<br>parts.nsfBy+Name?OpenView&Start=3                       |
|                                                       | Count = <i>n</i> , where <i>n</i> is the number of<br>rows to display                          | http://www.steve-o.com/parts/By+Name?<br>OpenView&Count=15                         |
|                                                       | ExpandView                                                                                     | http://www.steve-o.com/parts.nsf/By+Name?<br>OpenView&ExpandView                   |
|                                                       | CollapseView                                                                                   | http://www.steve-o.com/parts.nsf/By+Name?<br>OpenView&CollapseView                 |
|                                                       | Expand = <i>n</i> , where <i>n</i> is the row<br>to expand.                                    | http://www.steve-o.com/parts.nsf/By+Name?<br>OpenView&Collapse=3                   |
|                                                       | Collapse = <i>n</i> , where <i>n</i> is the row<br>to collapse.                                | http://www.steve-o.com/parts.nsf/By+Name?<br>OpenView&Collapse=7                   |
| OpenAbout (opens<br>the database's About<br>document) | http://Host/Database/\$about?<br>OpenAbout                                                     | http://www.steve-o.com/parts.nsf/\$about?<br>OpenAbout                             |
| OpenHelp (opens<br>the database's Help<br>document)   | http://Host/Database/\$help?<br>OpenHelp                                                       | http://www.steve-o.com/discussion.nsf/\$help?<br>OpenHelp                          |

| Action                                                                                                                        | Syntax                                                                                                     | Example                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| OpenIcon (opens the database's icon)                                                                                          | http://Host/Database/\$icon?<br>OpenIcon                                                                   | http://www.steve-o.com/homer+page.nsf/<br>\$icon?OpenIcon                                                    |
| Login argument (Add the Login argument at the end of <i>any</i> URL to force the user to enter their user name and password.) | http://Host/?OpenServer&login<br>http://Host/DatabaseDirectory/<br>DatabaseFileName?<br>OpenDatabase&login | http://www.steve-o.com/?<br>OpenServer&login<br>http://www.steve-o.com/<br>discussion.nsf?OpenDatabase&login |

## OPENING FORMS, NAVIGATORS, AND AGENTS

The following commands open Forms, Navigators, and agents.

| Action            | Syntax                                                                                                                                                                                                                                                                        | Example                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| OpenForm          | http://Host/Database/FormName?<br>OpenForm                                                                                                                                                                                                                                    | http://www.steve-o.com/products.nsf/Product?<br>OpenForm                                                                              |
|                   | http://Host/Database/<br>FormUniversalID?OpenForm                                                                                                                                                                                                                             | http://www.steve-o.com/products.nsf/<br>625E6111C597A11B852563DD00724CC2?<br>OpenForm                                                 |
|                   | http://Host/Database/FormNotelID?<br>OpenForm                                                                                                                                                                                                                                 | http://www.steve-o.com/<br>products.nsf/0000024A?OpenForm                                                                             |
|                   | http://Host/Database/\$defaultform?<br>OpenForm                                                                                                                                                                                                                               | http://www.steve-o.com/<br>products.nsf/\$defaultform?OpenForm                                                                        |
| Optional Argument | http://Host/Database/FormUniversalID?<br>OpenForm&ParentUNID, where<br>ParentUNID equals the Universal ID<br>of the parent document which is used<br>both in response Forms, and/or when<br>the Form's properties include "Formulas<br>inherit values from selected document" | http://www.steve-o.com/products.nsf/<br>27aa83d22c1e2c2227296262055df0e0?<br>OpenForm&ParentUNID=<br>8bc21a71614fd7ef852263de001f4a25 |
| OpenNavigator     | http://Host/Database/NavigatorName?<br>OpenNavigator                                                                                                                                                                                                                          | http://www.steve-o.com/products.nsf/<br>Main+Navigator?OpenNavigator                                                                  |
|                   | http://Host/Database/<br>NavigatorUniversalID?OpenNavigator                                                                                                                                                                                                                   | http://www.steve-o.com/products.nsf/<br>7B5BC17C7DC9EB7E85256207004F8862?<br>OpenNavigator                                            |
|                   | http://Host/Database/NavigatorNotelID?<br>OpenNavigator                                                                                                                                                                                                                       | http://www.steve-o.com/<br>products.nsf/000001AA?OpenNavigator                                                                        |
|                   | http://Host/Database/\$defaultNav?<br>OpenNavigator                                                                                                                                                                                                                           | http://www.steve-o.com/products.nsf/<br>\$defaultNav?OpenNavigator                                                                    |

| Action    | Syntax                                              | Example                                                                                |
|-----------|-----------------------------------------------------|----------------------------------------------------------------------------------------|
| OpenAgent | http://Host/Database/Agentname?<br>OpenAgent        | http://www.steve-o.com/sales/<br>leads.nsf/Process+New+Leads?OpenAgent                 |
|           | http://Host/Database/AgentUniversalID?<br>OpenAgent | http://www.steve-o.com/products.nsf/<br>D27E9F215AA6BDE585256324005D5F5D?<br>OpenAgent |
|           | http://Host/Database/AgentNoteID?<br>OpenAgent      | http://www.steve-o.com/<br>products.nsf/00230421B?OpenAgent                            |

## OPENING, EDITING, AND DELETING DOCUMENTS

The following commands manipulate documents.

| Action         | Syntax                                                                                                                                          | Example                                                                                                                        |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| OpenDocument   | http://Host/Database/View/<br>DocumentKey?OpenDocument<br><br>(DocumentKey is the contents of the<br>first sorted column in the specified View) | http://www.steve-o.com/products.nsf/<br>By+Name/PC213?OpenDocument                                                             |
|                | http://Host/Database/View/<br>DocumentUniversalID?OpenDocument                                                                                  | http://www.steve-o.com/products.nsf/By+Name/<br>21AE2FBFA573336A853468D100741784?<br>OpenDocument                              |
|                | http://Host/Database/View/<br>DocumentNoteID?OpenDocument                                                                                       | http://www.steve-o.com/leads.nsf/<br>By+Region/0000094E?OpenDocument                                                           |
| EditDocument   | http://Host/Database/View/Document/<br>?EditDocument                                                                                            | http://www.steve-o.com/products.nsf/<br>By+Name/?EditDocument                                                                  |
| DeleteDocument | http://Host/Database/View/Document?<br>DeleteDocument                                                                                           | http://www.steve-o.com/products.nsf/<br>By+Namer/?DeleteDocument                                                               |
| CreateDocument | http://Host/Database/Form/<br>?CreateDocument                                                                                                   | http://www.steve-o.com/products.nsf/<br>b9815a87b36a85d9852563df004a9533?<br>CreateDocument                                    |
| SaveDocument   | http://Host/Database/View/Document?<br>SaveDocument                                                                                             | http://www.steve-o.com/products.nsf/<br>a0cefa69d38ad9ed8525631b006582d0/<br>4c95c7c6700160e3650901df0078cfef?<br>SaveDocument |

## OPENING DOCUMENTS BY KEY

To open a document by key, create a View that sorts the documents by the first column, then use a URL to open the document based on the contents of the first column in the View.

The syntax for this URL is: **http://Host/DatabaseName/View/DocumentName?OpenDocument**, where **View** is the name of the View containig the document, and **DocumentName** is the key that is in the first sorted column of the View.

For example, the Domino URL *http://www.steve-o.com/register.nsf/New+Users/Laura+Lackner?OpenDocument* opens Laura's document from the New Users View in the database named `register.nsf`.

### Using @Commands to Link to a Document

Create a Link to a document in a View by key by combining **@Command([OpenView];"ViewName";"key")** with **@Command([OpenDocument])**, or **@Command([FileOpenDatabase];"";"DatabaseName";ViewName";"key")** with **@Command([OpenDocument])**.

For example, imagine an Action Hotspot on a Form that includes the combined formula **@Command([OpenView];"New Users";@UserName); @Command([OpenDocument])**. This would create a Link to a particular user's document in the New User View of a database. Since the *key* is the function **@UserName**, the target of the Link will change for each persopn opening the document.

Further refine the **@Command** with an optional argument for **[OpenDocument]**: "1" to edit, "0" to read.

## OPENING IMAGE FILES, ATTACHMENTS, AND OLE OBJECTS

The following command opens a file attachment from within a document.

| Action      | Syntax                                                                       | Example                                                                                         |
|-------------|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| OpenElement | <code>http://Host/Database/View/Document/ \$File/Filename?OpenElement</code> | <code>http://www.steve-o.com/products.nsf/ By+Name/OP223/\$File/chapter.txt? OpenElement</code> |



The following command opens an image from within a document.

| Action      | Syntax                                                                                                       | Example                                                                                                                                                                                                                                                                                                                        |
|-------------|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OpenElement | http://Host/Database/View/Document/<br>FieldName/FieldOffset?<br>OpenElement&FieldElemFormat=<br>ImageFormat | FieldOffset is represented by <i>xx.yy</i> , where <i>xx</i> is the field number, and <i>yy</i> is the byte offset into the field. ImageFormat is either GIF or JPEG. If the FileElemFormat is not entered, Domino assumes the image file format is GIF. (It is unlikely that that you will manually create this type of URL.) |

The following command opens an OLE object from within a document.

| Action      | Syntax                                                                                                        | Note                                                                  |
|-------------|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| OpenElement | http://Host/Database/View/Document/<br>FieldName/FieldOffset/\$OLEOBJINFO/<br>FieldOffset/obj.ods?OpenElement | (It is unlikely that that you will manually create this type of URL.) |

## SEARCHING FOR TEXT WITH DOMINO SEARCH URLs

The following commands search a Domino site or database.

| Action     | Syntax                                                                                                                                                                                                                                                                                                                    | Example                                                                                                                                                                 |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SearchSite | http://Host/Database/[\$SearchForm]?<br>SearchSite[ArgumentList]<br><br>(\$SearchForm and ArgumentList<br>are optional arguments; \$SearchForm<br>indicates that Domino will present a<br>search site Form for search input.<br>When this identifier is included in<br>a SearchSite URL, the ArgumentList<br>is ignored.) | http://www.steve-o.com/srchsite.nsf/<br>\$SearchForm?SearchSite<br><br>http://www.steve-o.com/srchsite.nsf/<br>?SearchSite&Query=new+product+<br>information;l;;0;FALSE |
| SearchView | http://Host/Database/View/<br>[\$SearchForm]?SearchView<br>[ArgumentList]                                                                                                                                                                                                                                                 | http://www.steve-o.com/products.nsf/<br>By+Name/\$SearchForm?SearchView<br><br>http://www.steve-o.com/products.nsf/<br>By+Name/?SearchView&Query=<br>PC156;3;;0;TRUE;   |

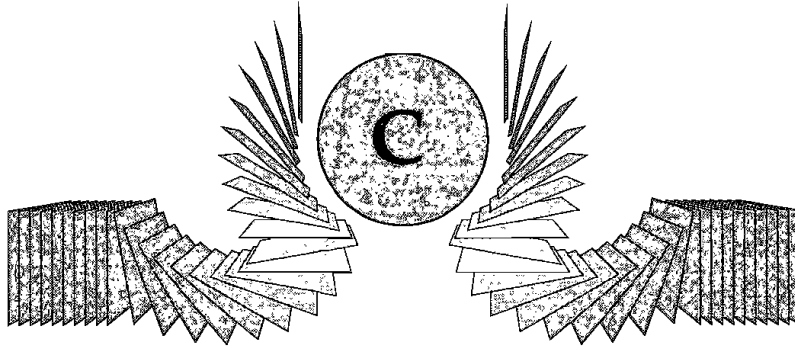
| Action             | Syntax                                                                                                                                                                                                                                                                                                                                                                           | Example                                                                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Optional Arguments | <p>ArgumentList= Query; SearchOrder;<br/>SearchThesaurus;SearchMax;SearchWV</p> <p>SearchOrder=[1,2,3] default = 1. 1 =<br/>"By Relevance", 2 = "By Date<br/>Ascending", 3 ="By Date Descending"</p> <p>SearchThesaurus=[TRUE, FALSE],<br/>default = FALSE</p> <p>SearchMax=[n], 0 default= 0<br/>(meaning all)</p> <p>SearchWordVariants=<br/>[TRUE, FALSE], default = TRUE</p> | <p>Where Query = the search string ;<br/>ArgumentList must contain the query<br/>argument—it can also contain any or all of<br/>the other arguments</p> |

## THE DOMINO URLs

| NotesObjectPath   | [? Action [Arguments]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NotesObjectPath = | <p>Server</p> <p>Server/Database</p> <p>Server/Database/View</p> <p>Server/Database/Form</p> <p>Server/Database/Navigator</p> <p>Server/Database/Agent</p> <p>Server/Database/View/Document</p> <p>Server/Database/View/Document/Field/Offset</p> <p>Server/Database/View/Document/Field/Offset/SubOffset</p> <p>Server/Database/DatabasesIcon</p> <p>Server/Database/DatabasesHelp</p> <p>Server/Database/DatabasesAbout</p> <p>Server/Database/Form/Field/Offset</p> <p>Server/Database/Form/Field/Offset/SubOffset</p> <p>Server/Database/Navigator/Field/Offset</p> <p>Server/Database/IconNote/Field/Offset</p> <p>Server/Database/HelpNote/Field/Offset</p> <p>Server/Database/AboutNote/Field/Offset</p> <p>Server/Database/View/\$SearchForm</p> |

| NotesObjectPath | [? Action [Arguments]]                                                                                                                                                                                                                                 |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NotesObject =   | Server<br>Database<br>View<br>Form<br>Navigator<br>Agent<br>Document<br>Field<br>Offset (within Field)<br>SubOffset (for example, attachment name)<br>Databaselcon<br>DatabaseHelp<br>DatabaseAbout<br>IconNote<br>HelpNote<br>AboutNote<br>SearchForm |
| Server =        | Nothing                                                                                                                                                                                                                                                |
| Database =      | /directory/filename.nsf                                                                                                                                                                                                                                |
| View =          | NotelD   NoteUNID   Name   \$defaultView                                                                                                                                                                                                               |
| Form =          | NotelD   NoteUNID   Name   \$defaultForm                                                                                                                                                                                                               |
| Navigator =     | NotelD   NoteUNID   Name   \$defaultNav                                                                                                                                                                                                                |
| Agent =         | NotelD   NoteUNID   Name                                                                                                                                                                                                                               |
| Document =      | NotelD   NoteUNID                                                                                                                                                                                                                                      |
| Field =         | Name                                                                                                                                                                                                                                                   |
| Offset =        | x.y                                                                                                                                                                                                                                                    |
| SubOffset =     | Name                                                                                                                                                                                                                                                   |
| Databaselcon =  | \$icon                                                                                                                                                                                                                                                 |
| DatabaseHelp =  | \$help                                                                                                                                                                                                                                                 |
| DatabaseAbout = | \$about                                                                                                                                                                                                                                                |
| IconNote =      | NotelD                                                                                                                                                                                                                                                 |
| HelpNote =      | NotelD                                                                                                                                                                                                                                                 |
| AboutNote =     | NotelD                                                                                                                                                                                                                                                 |
| SearchForm =    | \$SearchForm                                                                                                                                                                                                                                           |
| Action =        | Nothing   ImplicitAction   ExplicitAction                                                                                                                                                                                                              |

| NotesObjectPath  | [? Action [Arguments]]                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ImplicitAction = | Open<br>Create (supported on Documents only)<br>Edit (supported on Documents only)<br>Save (supported on Documents only)<br>Delete (supported on Documents only)<br>Search (supported on Views only)<br>Process (supported on Forms only)<br>OpenServer                                                                                                                                                                                         |
| ExplicitAction = | OpenDatabase<br>OpenView<br>OpenForm<br>OpenNavigator<br>OpenAgent<br>OpenDocument<br>OpenElement (bitmaps, attachments, and so on)<br>CreateDocument (Form Post Action only)<br>EditDocument<br>SaveDocument (Form Post Action only)<br>DeleteDocument<br>SearchSite<br>SearchView<br>ProcessForm (Form Post Action only)                                                                                                                      |
| Arguments =      | Start (OpenView)<br>Count (OpenView)<br>Expand (OpenView)<br>ExpandView (OpenView)<br>Collapse (OpenView)<br>CollapseView (OpenView)<br>FieldElemType (OpenElement)<br>FieldElemFormat (OpenElement)<br>SearchForm (Open the search Form)<br>Query (SearchSite and SearchView)<br>OldSearchQuery—which action?<br>SearchMax (SearchSite and SearchView)<br>SearchWV (SearchSite and SearchView)<br>SearchOrder<br>SearchThesaurus<br>ParentUNID |



# **DOMINO QUICK REFERENCE**

## **CREATING LINKS**

There are five ways to create a Link in Notes that will be translated to a Hypertext Link when the document, Form, View, or Navigator is displayed on the Web.

| Type of Link                     | Created by...                                                                                                                                                                                                                     | Notes                                                                                                                                                                              |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Notes Doc/View/<br>Database Link | Selecting the Link's target and choosing <b>Edit+Copy As Link+DocLink/View Link/Database Link</b> and then using <b>Edit+Paste</b> to insert it into a document or onto a Form.                                                   | This type of Link is represented by an icon on the Web page—the same one that is used to represent the Link in Notes. This type of Link can not be used on or to open a Navigator. |
| Link Hotspot                     | Selecting the Link's target, choosing <b>Edit+Copy As Link+DocLink/View Link/Database Link</b> and then selecting the text or graphic that will be the Link's Anchor on the Web and choosing <b>Create+Hotspot+Link Hotspot</b> . | Can only forge a Link to a Notes document, View, or database; can not be used to create a Link to another Web site or HTML-based Web page.                                         |
| Action Hotspot                   | Selecting the text or graphic that will be the Link's Anchor on the Web and choosing <b>Create+Hotspot+Action Hotspot</b> and then writing an <b>@URLOpen</b> formula for the Link's target.                                      | Can forge a Link to any Notes document, View, or database, <i>and</i> to any other Web site or HTML-based Web page.                                                                |
| Action Bar Button                | Writing a formula for the Action bar's Button                                                                                                                                                                                     | Can forge a Link to any Notes document, View, or database, <i>and</i> to any other Web site or HTML-based Web page.                                                                |
| HTML                             | By enclosing the <code>&lt;A HREF&gt;</code> HTML tag in [square braces]                                                                                                                                                          | Can forge a Link to any Notes document, View, or database, <i>and</i> to any other Web site or HTML-based Web page; <b>can also be used to display a Link in a View's column.</b>  |

## THE \$\$FIELDS AND \$\$FORMS

Domino uses special fields and Forms, all of which are identifiable by the fact that they begin with two dollar signs—as in `$$ViewBody` and `$$HTMLHead`. What, exactly, the \$\$ field or Form does depends very much on the context in which it is used, but in most cases these special fields and Forms are of no use to a Notes workstation user. For example, the `$$ViewBody` field will display the contents of a View on a Form that is displayed on the Web—but this field doesn't do anything when the same document is viewed in Notes.

### The \$\$Fields are:

Each of these fields is intended to be a text field. Since they are of little use to Notes workstation users, these fields are usually hidden when a Form is opened in Notes.

| Field name         | What It Does                                                                                            | How To Use It                                                                                                                                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$\$ViewBody       | Displays the contents of a View in a Form when the Form is displayed on the Web.                        | Write a formula for the field that returns the name of the View that is to be displayed. A Form can not have more than one \$\$ViewBody field.                                                                          |
| \$\$ViewList       | Displays the names of the database's Views and Folders in a Form when the Form is displayed on the Web. | This field does not need—and will ignore—any formula you write for it.                                                                                                                                                  |
| \$\$NavigatorBody  | Displays the contents of a Navigator in a Form when the Form is displayed on the Web.                   | Write a formula for the field that returns the name of the Navigator that is to be displayed. If a Form is to include multiple \$\$NavigatorBody fields, name them \$\$NavigatorBody_1, \$\$NavigatorBody_2, and so on. |
| \$\$Return         | Creates a special Web page that is shown to a Web user after saving a Form in a database.               | Write a formula for the field that includes the text and HTML tags that you want to use to create the page.                                                                                                             |
| \$\$HTMLHead       | Constructs a Web page's Head tag.                                                                       | Write a formula that returns a text value containing the HTML, Meta tags, or JavaScript that should be included between the <HEAD> and </HEAD> tags for the Web page.                                                   |
| \$\$QueryOpenAgent | Runs an agent before a document is converted to HTML and displayed to the Web user.                     | Write a formula for the field that returns a text value that equals the name of the agent that should be run.                                                                                                           |
| \$\$QuerySaveAgent | Runs an agent before a document that was composed by a Web user is saved to disk on the Domino server.  | Write a formula for the field that returns a text value that returns the name of the agent that should be run.                                                                                                          |

### The \$\$Forms are:

Each of these Forms is used to display the contents of a View, Navigator, or search page when the View, Navigator, or search page is displayed on the Web. Using these Forms lets you fully customize the way the View, Navigator or search is displayed.

| Form name                                  | What It Does                                                                                                                                   | How To Use It                                                                                                                                                                                                                                                                     |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\$\$ViewTemplate for N</code>        | Displays the contents of the View named (or with the name alias) <i>N</i> , only when the View is displayed on the Web.                        | The Form must include, at the very least, a field named <code>\$\$ViewBody</code> . This field tells Domino where to display the documents that are in View <i>N</i> . <code>ViewTemplate</code> Forms can usually include other fields, images, text, HTML tags, and Navigators. |
| <code>\$\$Viewtemplate Default</code>      | Displays the contents of any View that does not have its own <code>\$\$ViewTemplate for N</code> Form.                                         | This Form also must include a <code>\$\$ViewBody</code> field.                                                                                                                                                                                                                    |
| <code>\$\$NavigatorTemplate for N</code>   | Displays the Navigator named <i>N</i> , only when the Navigator is displayed on the Web.                                                       | <code>NavigatorTemplate</code> Forms usually include other fields, images, text, HTML tags, and Navigators.                                                                                                                                                                       |
| <code>\$\$NavigatorTemplate Default</code> | Displays the contents of any Navigator that does not have its own <code>\$\$NavigatorTemplate for N</code> Form.                               |                                                                                                                                                                                                                                                                                   |
| <code>\$\$SearchTemplate for N</code>      | Similar to a <code>ViewTemplate</code> Form, this Form displays the results of a Full-Text Search on the View named <i>N</i> .                 | The Form must include a <code>\$\$ViewBody</code> field. <code>SearchTemplate</code> Forms can also include other fields, images, text, HTML tags, and Navigators.                                                                                                                |
| <code>\$\$SearchTemplateDefault</code>     | Used to display the results of a Full-Text Search performed on any View that does not have its own <code>\$\$SearchTemplate for N</code> Form. | The Form must include a <code>\$\$ViewBody</code> field.                                                                                                                                                                                                                          |
| <code>\$\$Search (alias)</code>            | Used on the Web to gather the criteria upon which a search will be performed.                                                                  |                                                                                                                                                                                                                                                                                   |

## Special Actions For Views and Search Results that Are Displayed on the Web Through a Form

When a Notes View (that does not have a `$$ViewTemplate` Form) is displayed on the Web, Domino automatically includes Action Hotspots that the Web user uses to navigate to the next or previous part of the View, to perform a search, and so on. Since these Action Hotspots are not automatically added to Views and search results that are displayed on the Web through one of the aforementioned `$$Forms`, you must create Action Hotspots on the `$$form`—or include create an Action bar button for the View—with each of the following:



| Suggested Action Name | Formula                                 |
|-----------------------|-----------------------------------------|
| Next                  | @DbCommand("Domino";"ViewNextPage")     |
| Previous              | @DbCommand("Domino";"ViewPreviousPage") |
| Expand                | @Command([ViewExpandAll])               |
| Collapse              | @Command([ViewCollapseAll])             |
| Search                | @Command([ViewShowSearchBar])           |

(Instead of text, each of these formulas is often associated with an icon or image on the Form, or is included in a Navigator that is displayed on the \$\$form through a \$\$NavigatorBody field.)

## HOW NOTES PROPERTIES WORK ON THE WEB

### Database Launch Properties

| Property                        | Supported? | Notes                                                                                                                                 |
|---------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| On Database Open Launch options | Yes        | (1) Except for "Launch 1st attachment" in "About Database." (2) Use a database's Launch Property to set the Home Page for a database. |

### Form Properties

| Property                                                  | Supported ? | Notes                                                                                                                                                                                                                     |
|-----------------------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Merge replication conflicts                               | No          | By definition, Replication Conflicts can't originate from Web users.                                                                                                                                                      |
| Version control                                           | No          |                                                                                                                                                                                                                           |
| Anonymous Forms                                           | No          | Giving Unauthenticated users the ability to compose documents in a database simulates this feature.                                                                                                                       |
| Automatically refresh fields                              | No          |                                                                                                                                                                                                                           |
| Disable field exchange                                    | No          |                                                                                                                                                                                                                           |
| On Create: Formulas inherit values from selected document | Yes         | (1) Except for inheriting a value into a Rich-Text field. (2) This only applies to a document that is created by a Web user who is viewing a document, since there is no way to select a document in a View (on the Web). |

| Property                                                         | Supported ? | Notes                                                                                                                                                                      |
|------------------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| On Create: Inherit entire document into selected Rich-Text field | No          |                                                                                                                                                                            |
| On Open: Show context pane                                       | No          |                                                                                                                                                                            |
| Store Form in document                                           | Yes         | If a document was created by a Notes user, it will be displayed through a stored Form on the Web. Web users can not, however, store Forms in documents when creating them. |
| Autolaunch options (1st attachment, 1st OLE object, 1st DocLink) | No          |                                                                                                                                                                            |
| Default encryption keys                                          | No          | Web users can not encrypt/decrypt field values.                                                                                                                            |
| Disable printing/ forwarding/ copying from Clipboard             | No          |                                                                                                                                                                            |

## The Parts of a Form

| Form Part                               | Supported? | Notes                                                                                                                                                                            |
|-----------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attachments                             | Yes        | For a Web user to be able to upload an attachment, Web Forms need a special Hotspot. A Web user can download any type of attachment.                                             |
| Buttons                                 | No         | A Web Form can have only one button, and it is <i>always</i> used as a Submit button.                                                                                            |
| Graphics                                | Yes        |                                                                                                                                                                                  |
| Hotspots                                | Yes        | Except for Popups.                                                                                                                                                               |
| Layout regions                          | No         |                                                                                                                                                                                  |
| ActiveX components, OLE and OCX objects | Yes        | Saving edits to objects and components is not supported on the Web.                                                                                                              |
| Tables                                  | Yes        | Domino displays all of the cells in a table with borders; if the table's top left cell has a border, the entire table is displayed with a border; otherwise, there is no border. |

## Field Properties

| Property/Attribute            | Supported?     | Notes                                                                                                                   |
|-------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------|
| Keyword fields                | Yes            | Control the number of choices visible in the field by using the <code>&lt;SIZE=n&gt;</code> HTML tag in the field help. |
| Name fields                   | Yes            | Except for Use Address dialog for choices, Use View dialog for choices, and Use Access Control List for choices.        |
| Computed-when-composed fields | Yes            | Don't use formulas based on time computations.                                                                          |
| Notes/FX fields               | No             |                                                                                                                         |
| Default value formulas        | Yes            |                                                                                                                         |
| Compute after validation      | No             | Use a <b>\$\$QuerySaveAgent</b> to compute values after a document has been created.                                    |
| Field help                    | Not supported. | The Field input box is used by Domino for HTML tags that apply to the field, such as <code>&lt;SIZE&gt;</code>          |
| Field-level encryption        | No             | Web users can not encrypt/decrypt field values.                                                                         |
| Give this field default focus | No             |                                                                                                                         |
| Signed fields                 | No             |                                                                                                                         |

## View Properties

| Property/Attribute                             | Supported? | Notes                                                                                                                                                  |
|------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Calendar-style Views                           | Yes        | Except for conflict bars, the clock, creating new appointments, and scrolling through entries within a single day.                                     |
| Folders                                        | Yes        | Web users can not move documents into Folders.                                                                                                         |
| Multi-line column headings and multi-line rows | Yes        | To prevent line wrap, enter 1 in the Lines per heading field in the View's Properties box. The same guidelines are true for the Lines per row setting. |
| On Open, Go To options                         | No         |                                                                                                                                                        |
| On Refresh options                             | No         |                                                                                                                                                        |
| Show in View menu                              | No         |                                                                                                                                                        |

| Property/Attribute                                                                                      | Supported? | Notes                                                       |
|---------------------------------------------------------------------------------------------------------|------------|-------------------------------------------------------------|
| Style options for Unread rows, Alternate row colors, Show selection margin, and Beveled column headings | No         |                                                             |
| View Indexing options                                                                                   | No         |                                                             |
| Click on column header to sort                                                                          | No         |                                                             |
| Collapsed/expanded categories                                                                           | Yes        |                                                             |
| Resizable column                                                                                        | No         |                                                             |
| "Show twisties" when row is expandable                                                                  | Yes        | In fact, twisties are <i>always</i> displayed in Web Views. |

## Navigator Properties

If you include an unsupported objects in a Navigator, the object's bitmap will be displayed on the Web, but it will not work.

| Property/Feature                                | Supported? | Notes                                              |
|-------------------------------------------------|------------|----------------------------------------------------|
| Auto adjust panes at runtime                    | No         |                                                    |
| Highlight when... options for Navigator objects | No         |                                                    |
| Navigator graphics                              | Yes        | Only Graphic backgrounds are supported.            |
| Navigator objects                               | Yes        | Only Hotspot polygons and Hotspot rectangles.      |
| Alias a Folder Hotspot action                   | No         | Dragging and dropping to Folders is not supported. |

## Actions, Agents, and Scripts

| Property/Feature                                                                   | Supported? | Notes                                                                                                                                                              |
|------------------------------------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actions from Actions menu                                                          | No         | Web users don't have an Actions menu.                                                                                                                              |
| Agents                                                                             | Yes        | Except for "If Document Has Been Pasted" and "Selected documents."                                                                                                 |
| LotusScript for Forms, Actions, and buttons                                        | No         | Use \$\$Return field and CGI scripts to run programs or a \$\$QuerySaveAgent field or \$\$QueryOpenAgent field to run agents before a document is saved or opened. |
| Simple Actions for agents and Form and View Actions                                | No         | You must write formulas to emulate the desired simple Action.                                                                                                      |
| System Actions supplied with Forms and Views (such as *Edit Document, *Categorize) | No         | You must write formulas to emulate the desired simple Action.                                                                                                      |

## @Functions

| @Function                   | Supported? | Notes                                                                                                                                       |
|-----------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| @Certificate                | No         |                                                                                                                                             |
| @DbCommand                  | Yes        | Only @DbCommand ("Domino";"ViewNextPage") and @DbCommand ("Domino";ViewPreviousPage") to create a Link to the next/previous page in a View. |
| @DDEExecute                 | No         |                                                                                                                                             |
| @DDEInitiate                |            |                                                                                                                                             |
| @DDEPoke                    |            |                                                                                                                                             |
| @DDETerminate               |            |                                                                                                                                             |
| @DocMark                    | No         |                                                                                                                                             |
| @DeleteDocument             | No         |                                                                                                                                             |
| @DocChildren                | Yes        | Only in View and column formulas.                                                                                                           |
| @DocDescendants             |            |                                                                                                                                             |
| @DocLevel                   |            |                                                                                                                                             |
| @DocNumber                  |            |                                                                                                                                             |
| @DocParentNumber            |            |                                                                                                                                             |
| @DocSiblings                |            |                                                                                                                                             |
| @IsCategory                 |            |                                                                                                                                             |
| @IsExpandable               |            |                                                                                                                                             |
| @Responses                  |            |                                                                                                                                             |
| @DialogBox                  | No         |                                                                                                                                             |
| @PickList                   |            |                                                                                                                                             |
| @Prompt                     |            |                                                                                                                                             |
| @IsModalHelp                |            |                                                                                                                                             |
| @GetPortsList               |            |                                                                                                                                             |
| @Environment                | No         | Use CGI variables.                                                                                                                          |
| @SetEnvironment             |            |                                                                                                                                             |
| ENVIRONMENT keyword         |            |                                                                                                                                             |
| @MailSend                   | No         |                                                                                                                                             |
| @Domain                     | No         |                                                                                                                                             |
| @MailDbName                 |            |                                                                                                                                             |
| @MailEncryptSavedPreference |            |                                                                                                                                             |
| @MailEncryptSendPreference  |            |                                                                                                                                             |
| @MailSavePreference         |            |                                                                                                                                             |
| @MailSignPreference         |            |                                                                                                                                             |
| @IsAgentEnabled             | No         |                                                                                                                                             |

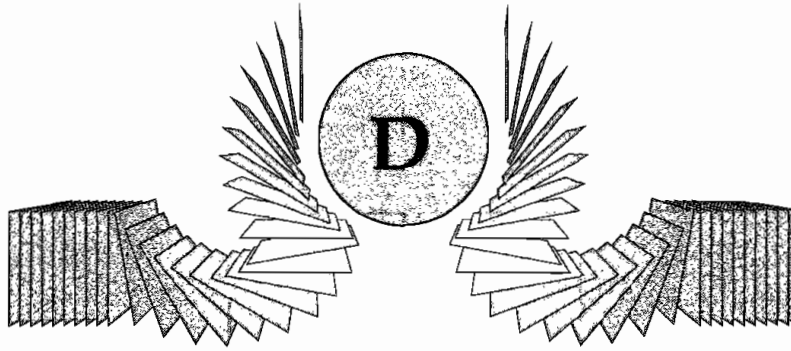
| @Function               | Supported? | Notes                                                                                                                                                         |
|-------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @IsDocBeingMailed       | No         |                                                                                                                                                               |
| @IsDocBeingRecalculated |            |                                                                                                                                                               |
| @Created                | Yes        | Values based on time computations in computed-when-composed fields, such as @Now and @DocumentUniqueID may be updated a second time during a Web transaction. |
| @Modified               |            |                                                                                                                                                               |
| @Now                    |            |                                                                                                                                                               |
| @Today                  |            |                                                                                                                                                               |
| @Tomorrow               |            |                                                                                                                                                               |
| @Yesterday              |            |                                                                                                                                                               |
| @DocumentUniqueID       |            |                                                                                                                                                               |
| @URLGetHeader           | No         |                                                                                                                                                               |
| @URLHistory             |            |                                                                                                                                                               |
| @UserPrivileges         | No         |                                                                                                                                                               |
| @UserRoles              | Yes.       | Appends \$\$WebClient to the list of roles.                                                                                                                   |
| @Platform               | Yes        |                                                                                                                                                               |

## @Commands

Only the following @Commands are supported on the Web.

| @Command                                                      | Notes                                                                                           |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| @Command([CalendarFormat])                                    |                                                                                                 |
| @Command([CalendarGoTo])                                      |                                                                                                 |
| @Command([Compose])                                           | The Server argument doesn't apply/work.                                                         |
| @Command([EditClear])                                         | Only for Forms; not in View Actions.                                                            |
| @Command([EditDocument])                                      | Only for Forms; not in View Actions.                                                            |
| @Command([EditInsertFileAttachment])                          | Used in an Action Hotspot to create the input field necessary for attaching files from the Web. |
| @Command([FileOpenDatabase])                                  | The server argument must be ""                                                                  |
| @Command([FileOpenDatabase];"":<br>"dbname";"ViewName";"key") | When followed by the <b>OpenDocument</b> command, allows you to specify the database name.      |
| @Command([NavigateNext])                                      | Only for Forms; not in View actions.                                                            |

| @Command                                  | Notes                                                                                                                |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| @Command([NavigatePrev])                  |                                                                                                                      |
| @Command([NavigateNextMain])              |                                                                                                                      |
| @Command([NavigatePrevMain])              |                                                                                                                      |
| @Command([OpenNavigator])                 |                                                                                                                      |
| @Command([OpenView])                      |                                                                                                                      |
| @Command([OpenView];<br>"ViewName";"key") | Must be followed by the <b>OpenDocument</b> command to produce a Domino URL that opens a specific document by a key. |
| @Command([OpenDocument])                  |                                                                                                                      |
| @Command([ToolsRunMacro])                 |                                                                                                                      |
| @Command([ViewChange])                    | Works like OpenView, but the View name must be included.                                                             |
| @Command([ViewExpandAll])                 |                                                                                                                      |
| @Command([ViewCollapseAll])               |                                                                                                                      |
| @Command([ViewShowSearchBar])             |                                                                                                                      |



# **HOSTING MULTIPLE WEB SITES ON A SINGLE DOMINO SERVER**

Of particular interest for administrators of Domino servers for Internet Service Provider or large corporate Intranets is the ability to create *Virtual servers*. When a single Domino server has been configured as such, it appears for all intents and purposes as different Web sites—even though these “sites” are all hosted on a single Domino server.

For example, an ISP providing Web services to many customers can assign each of the customers their own Web address—*www.company1.com*, *www.company2.com*, and so on—but run each of the sites on one computer. Running several sites on one system obviously results in less time spent administering the server, and reduced support costs for the machine.



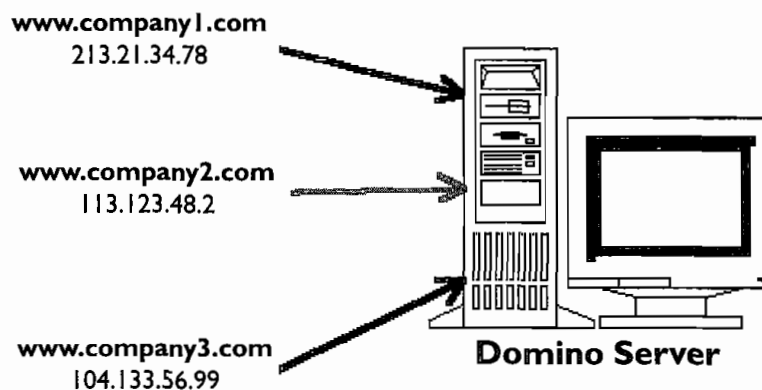


**Figure D.1** A Domino server can host multiple Virtual Domino servers—each with its own IP address and host name.

When a Domino server is configured to run several of these Virtual servers, each server gets its own domain name, IP address, and Home Page. If necessary, the Virtual Web servers can also be configured to use their own separate HTML, CGI, and Icon directories.

## THE DOMINO CONFIGURATION DATABASE

A Domino server that will be configured as more than one Virtual server must have a special database, typically known as Domino Configuration (`domcfg.nsf`). This database is used by the Domino server to configure the IP addresses of each of the Virtual servers, their default Home Pages, HTML directories, and so on.

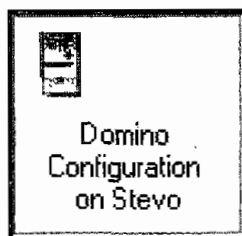


**Figure D.2** The Domino Configuration database is used to define the characteristics of Virtual Domino servers.

Unlike other administrative databases on the Domino server (like the N&A Book, Database Catalog, and so forth). This Configuration database is not created automatically, so the first step in setting up Virtual Domino servers is to manually create the server's Domino Configuration database.

To create a new Domino Configuration database, choose **File+Database+New**. Although the new database can have any title, its file name *must* be `domcfg.nsf`, and the new database's design must be based on the Domino Configuration template (`domcfg.ntf`).

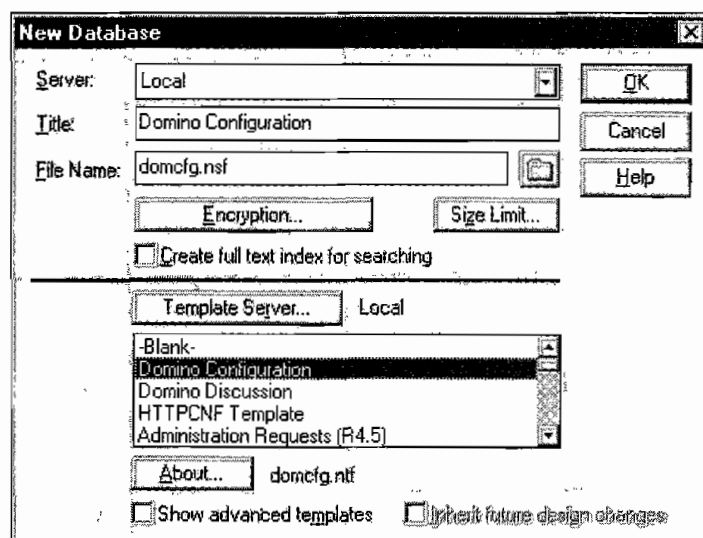


**Figure D.3** A new Domino Configuration database must be named `domcfg.nsf`.

## THE VIRTUAL SERVER CONFIGURATION DOCUMENT

Each Virtual server on a Domino server requires its own Virtual Server document in the Domino Configuration database; if the Domino server was to host six Virtual servers, there will be six Virtual server documents in the Configuration database.



**NOTE**

Configuration settings defined on the Virtual Server document in the Domino Configuration database override those defined on the Domino server's Server document in the Public N&A Book.

The Configuration document has nine fields—but the most important are the first and the fifth. The first field on the document (`IP Address`) is for the Virtual server's TCP/IP address. Each Virtual server must have its own, unique IP address; this IP address must be static, and you can not map multiple server names to the same IP address. In other words, if the Domino server was configured to host six Virtual servers, each Virtual server would have its own Virtual Server document in the Configuration database, and each document would have its own **unique** IP address. (And, presumably, each of these IP addresses would also be matched with a particular host name and registered with InterNIC, in the case of an Internet-based site.)

The fifth field (`Home URL`) tells the Domino server where to find the Home page for the Virtual server; in most cases, this field should contain the name of the database that contains the Virtual server's Home Page: `company1.nsf`, for example. If the Virtual server's Home page is instead based on an HTML file, the name of the file goes in the `Default home page` field.

The fields in the Mapping section of the Virtual Server document need only be used if the Virtual server's HTML, CGI, and Icon paths are different from the Domino server's defaults. Typically, several Virtual servers can share the same file locations and paths; but, if two of the Virtual servers needed access to different CGI scripts or HTML files that happened to have the same file names, the files would be stored in different directories on the Domino server, and the appropriate directory names would be entered into these fields. (Chapter 7 fully explains the use of each these directories.)

## Mapping Directories on a Domino Server

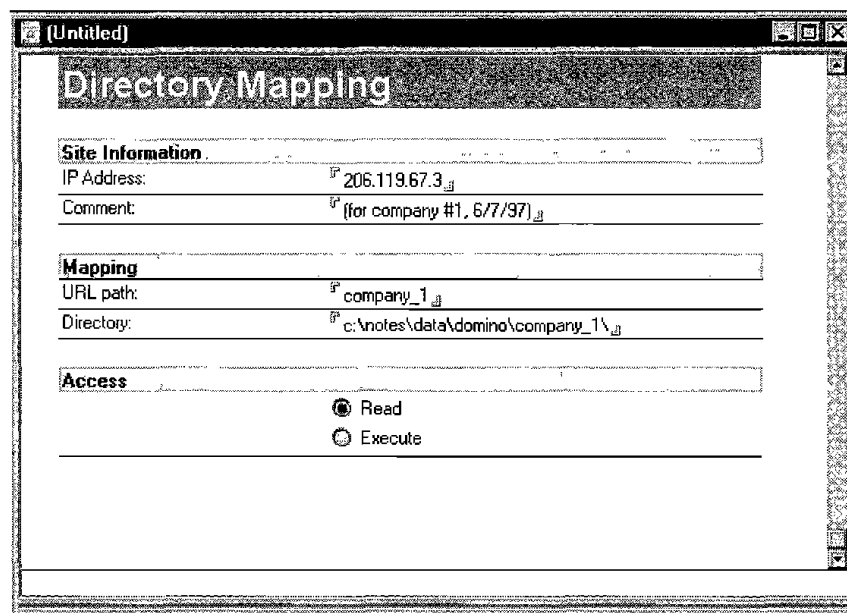
Directory Mapping lets the administrator configure directories on a Domino server to appear to be on the root directory of the server, even if they aren't. This feature is especially useful on a Virtual server, where the site can be configured so that a user requesting an image for Company1's Home Page (<http://www.company1.com/logo.gif>) might actually be served up a file that's located in the `c:/notes/data/domino/company1.com/icons` directory. Mapping directories frees a Web user from having to know the internal configuration of the server and makes the site's URLs easier to use and navigate.

To create a Directory Mapping on a Domino server, create a **Mapping URL -> Directory** document in the Domino Configuration database.

Enter the IP address of the server that will be configured in the **IP Address** field, the URL path being mapped in the **URL Path** field, and the actual directory name for those files in the **Directory** field.

For example, to indicate that the locations for the images in Company1's site, you might enter **/images** in the **URL Path** field, and **c:** in the **Directory** field.

The **Access** field is used to determine what rights the Web user has to the files in the directory that's being mapped. If the mapped directory contains contents files—such as HTML files, images, or Notes databases—assign **Reader** access to the directory. If the mapped directory contains CGI scripts or Perl programs that the user will run, grant **Execute** access.



The screenshot shows a web browser window titled "(Untitled)" displaying a "Directory Mapping" form. The form is divided into three sections: "Site Information", "Mapping", and "Access".

| Site Information |                          |
|------------------|--------------------------|
| IP Address:      | 206.119.67.3             |
| Comment:         | (for company #1, 6/7/97) |

| Mapping    |                                 |
|------------|---------------------------------|
| URL path:  | company_1                       |
| Directory: | c:\notes\data\domino\company_1\ |

| Access                           |         |
|----------------------------------|---------|
| <input checked="" type="radio"/> | Read    |
| <input type="radio"/>            | Execute |

**Figure D.4** Creating a Directory Mapping document in the Domino Configuration database.

## Mapping a URL to a Different Location

Mapping a URL is a way to accomplish a couple of different tasks. Imagine a Web site that has been up for a few months, with plenty of

visitors. Some of these visitors will create Bookmarks to the pages on the site; mapping an individual URL is an easy way to move files on a Domino server to a new location (perhaps a different directory, or to a different volume on the server altogether) without breaking the users' Bookmarks (happy users = frequent visitors, presumably). Mapping URLs also can be a easy way out of renaming directories, and can afford the opportunity to hide especially long directory and file names.

To create a Directory mapping on a Domino server, create a **Mapping URL-> URL** document in the Domino Configuration database. Enter the IP address of the server that will be configured in the IP Address field, the "old" URL path in the First URL Path field, and the new location in the file in the Second URL Path field.

For example, imagine that the description of Company1's Products used to be retrieved via *http://www.company1.com/products*, and that the page associated with that URL was located in *c:*. For the sake of the argument, we'll say the page has been moved to a new directory/volume on the server: *d:*. To preserve any Bookmarks or links that users might have to the page, the site would be configured as shown in Figure D.5.

| URL Mapping             |                               |
|-------------------------|-------------------------------|
| <b>Site Information</b> |                               |
| IP Address:             | 199.32.2104.66                |
| Comment:                | [for company #1 - 7/1/97]     |
| <b>Mapping</b>          |                               |
| First URL path:         | c:\notes\data\domino\company1 |
| Second URL path:        | d:\company1                   |

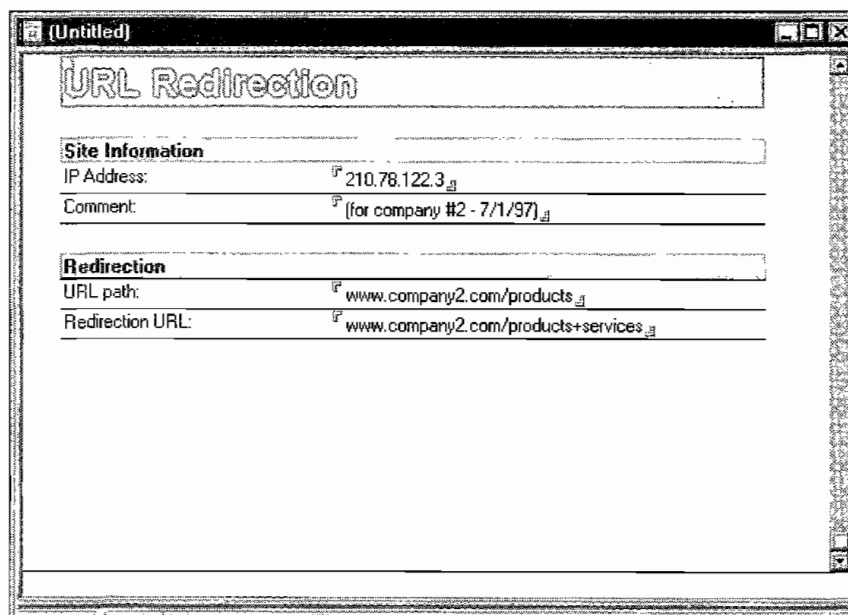
Directory in file system where CGI Programs are stored

**Figure D.5** A sample Mapping URL -> URL document in the Domino Configuration database.

## Redirecting URLs

Redirecting URLs lets the administrator configure the Domino server to serve up the page at URL "X" when a user requests the page at URL "Y." This feature is useful if a particular Web site needs to be moved to a different Domino server.

To configure a redirected URL, create a **Redirection URL -> URL** document in the Domino Configuration database. Enter the IP address of the server that will be configured in the IP Address field, the URL that users will be requesting in the URL Path field, and enter the URL to which users should be sent instead in the Redirection URL Path field.



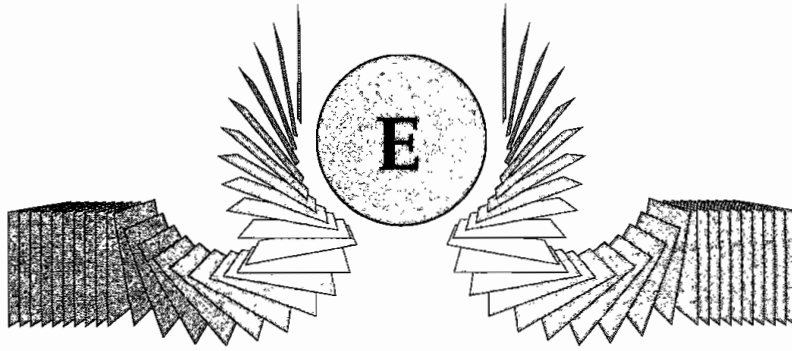
The screenshot shows a web browser window with the title '(Untitled)'. The main content area is titled 'URL Redirection'. Below the title, there are two sections: 'Site Information' and 'Redirection'. The 'Site Information' section contains two fields: 'IP Address' with the value '210.78.122.3' and 'Comment' with the value '(for company #2 - 7/1/97)'. The 'Redirection' section contains two fields: 'URL path' with the value 'www.company2.com/products' and 'Redirection URL' with the value 'www.company2.com/products+services'.

| Site Information |                           |
|------------------|---------------------------|
| IP Address:      | 210.78.122.3              |
| Comment:         | (for company #2 - 7/1/97) |

| Redirection      |                                    |
|------------------|------------------------------------|
| URL path:        | www.company2.com/products          |
| Redirection URL: | www.company2.com/products+services |

**Figure D.6** A sample Redirection URL -> URL document in the Domino Configuration database.



## **THIS JUST IN: THE LOTUS WEBLICATOR**

Just as we went to press with this fine tome, Lotus announced some exciting new technology, known as the *Weblicator*. It's too early for this edition of the book to provide much detail about how the product works—it's still being developed—but we can provide some background about this sure-to-be-a-winner program.

### **WHAT IT IS**

The Lotus Weblicator extends the best and brightest features of a Notes workstation, such as a local Object store, Views, Full-Text searching, Replication, and Security—to any Web browser program. This means that a user of, say, Netscape's Navigator can store the Web pages he has retrieved in a local database, which is perfect for off-line browsing. Since the Web pages are stored right on his workstation, a user could also use the Weblicator software to replicate (or *Weblicate*) the database to another computer, or perhaps create agents that search the pages they have retrieved for information.

The Weblicator also comes ready-made with the Web Ahead and Page Minder agents that can be used to monitor pages for changes and retrieve linked pages for off-line surfing.

## WHAT IT CAN DO

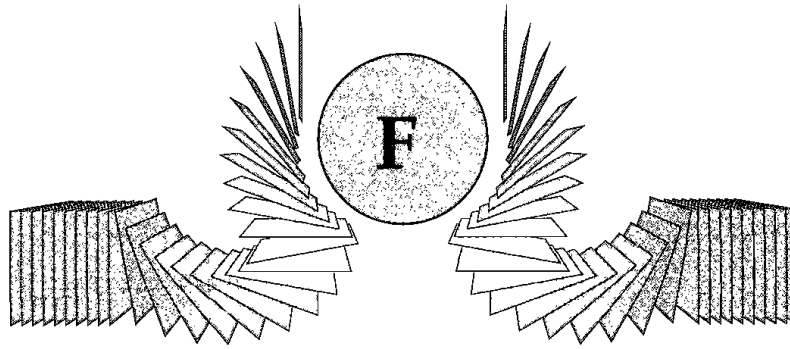
More specifically, the Weblicator extends the following functionality to any Web user:

- **Scheduled Weblication:** A user's workstation can be configured to Weblicate with any Web server to retrieve and store Web pages for local browsing, even when the workstation is not connected to the network.
- **The Web Ahead Agent:** Users can take advantage of the PageMinder agent that will pre-retrieve all or parts of a Web site for off-line surfing.
- **The PageMinder Agent:** Web users can also employ this agent to automatically re-retrieve Web pages that have changed since the last download.
- **A local Object store:** All of the pages that a user has retrieved are saved and stored in a database on his workstation, including text, images, Links, and Java applets or ActiveX components.
- **Database views:** As with any other database, the Views are used to organize and sort the Web pages that the user has retrieved.
- **Performance:** Since the Weblicator can be configured to retrieve pages in a "background" mode, users interact with the Web faster.
- **Security:** The Web pages stored locally can take advantage of Notes database encryption for added security, and are also governed and secured by a standard Lotus Notes Access Control List.
- **Cross Platform:** The Weblicator works with any Web browser program that supports frames, but, according to Lotus, is "optimized for Netscape Navigator and Microsoft Internet Explorer."

## WHERE TO GET MORE INFORMATION

At press time, a beta copy of the Weblicator was available for download at <http://beta.notes.net/Downwel.nsf/Web+DownloadSelector>. If that link has gone stale, more information can certainly be found at Lotus' Home Page, at <http://www.lotus.com>.





# USING THE SECURED SOCKETS LAYER METHOD OF SECURING WEB TRANSACTIONS

## WHY USE SECURED SOCKETS LAYER?

Secured Sockets Layer (SSL) provides a way for the Domino Webmaster and user to authenticate and “privatize” one another’s identity by using a mechanism of *Certificates* and *key files* similar to the USER.IDs and Certificates that are the basis of “regular” Notes security. More specifically, SSL offers these three features:

- **Message Encryption:** HTTP data being sent from the user to the server or the server to the user is encrypted while in transit. This is especially useful for sites involved in commerce with the public, as it allows the Domino Webmaster to allay the fears of potential customers by assuring them that important information—such as names, addresses, or even credit card numbers—is encrypted in such a manner that it can not be compromised.

- **Message Validation:** A special *message digest* is included in all traffic between the server and user. Functioning much like the traditional “checksum” that is used in data communications to ensure the validity of the communications, this message digest is used in SSL to detect if the data has been tampered with while enroute.
- **Server Authentication:** A special *digital signature* accompanies traffic being sent from the Domino server to the user—yet another way to guarantee to Web users and potential Web customers that the Web server “really is who it says it is.”

## HOW IT WORKS

When a Domino server and user use SSL, the transactions that pass between them are encrypted based on a public/private key scheme known as an *RSA-based cryptosystem*. This system is very similar to Notes workstation/server encryption, with just a few minor differences in vocabulary. In the world of SSL transactions, both the user and the server possess a special file, known as a *Key Ring*, that holds the user or server’s Public and Private Keys, as well as the Certificate that is used to check the authenticity of the data passing between them. The responsibility for the authentication and encryption falls upon a Certificate that is used and shared between the server and user—in other words, the Key Ring files located on both the Domino server and Web user’s workstations must contain the same Certificate.

For a public Web server, it is most common that the user and server would share a Certificate that came from a *Central Authority* (CA), such as VeriSign. VeriSign is a company that will—for a fee, of course—issue these Certificates to users and servers. (The Domino administrator can also create Certificates for use in Intranet applications where this level of security is required, or perhaps for testing and staging purposes.)

To access a site using SSL, the Web user employs a slightly different URL; rather than identifying the server with the *http://* Web protocol, the user enters *https://*. For example, to secure access to MIS:Press's Web site, the user would enter the URL *https://www.mispress.com*.

**NOTE**

Not all Web browsers support the use of SSL security. The good news is that Netscape's Navigator and Microsoft's Internet Explorer *do*.

When a user contacts a Web site by using the *https://* convention, the Domino server automatically uses the Certificate that the user and server share to encrypt the traffic between them. If the user does not have an SSL Certificate in common with the server, the Domino server leads the user "by the hand" through the process of obtaining the requisite Certificate.

## USING AND MANAGING SSL

Lotus provides a special SSL Administration database that is used to create and manage SSL for a Domino server.

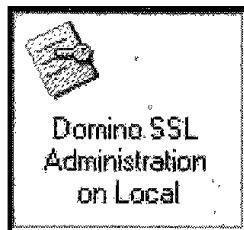


**Figure F.1** Use the SSL Administration database to create and manage SSL.

When you open this database, Notes displays its About document; click the picture of the keys to proceed.



**Figure F.2** Click the picture of the keys to manage SSL.



**NOTE**

Make sure you are accessing the database from the Domino server ("sitting at the server," in other words) when you use this database.

## Requisite Vocabulary

### The SSL Certificate Name

The SSL Certificate that is used to secure Web transactions includes a distinguished name, which is automatically based on either the server or CA's name and geographical location. The dialog boxes that Domino presents when you create a Certificate include fields for these names, as well as some other (optional) fields, such as *State* and *Country*. (It stands

to reason that the more fields you use to identify the Certificate, the fewer the chances that you will encounter another Certificate on the Web with that same name.) Specifically, the parts of a Certificate name include:

- **Common name:** which—for a Domino server—is the required fully-qualified Host name as listed in the Server document in the Public N&A Book. Example: *www.steve-o.com*. For a CA, the Certificate's name is the company's name. Example: VeriSign SSLCA or ACME.
- **Organizational Unit:** which is the optional group or department that the Certificate's owner belongs to, such as Sales.
- **Organization:** the required name of the organization that the Certificate's owner belongs to, which is usually the company's name.
- **City or Locality:** the optional municipality of the owner. Example: Boston.
- **State or Province:** at least three (required) characters that represent the owner's state.
- **Zip or Postal Code:** self-explanatory, and optional.
- **Country:** Two required characters representing the country where the Certificate owner resides.

#### The SSL Key Ring

An SSL Key Ring has three parts:

- **File Name:** which is the file name for the Key Ring that resides on each SSL server and user; the default for which is KEYFILE.KYR. If you use a different name, make sure to change the name of the file as it is listed in the *Security* section of the Server document for the Domino server in the Public N&A Book. Each SSL-enabled Domino server has a Key Ring file, which may contain more than one Certificate.
- **Key Name:** which is a case-sensitive name by which the Key is known.
- **Password:** which is at least six characters long and is case-sensitive. Anyone using the Key Ring must know this password.

## Enabling SSL Transactions on a Domino Server

To “turn on” SSL on a Domino server, set the **Accept SSL Site Certificates** field in the **Security** section of the Domino server’s Server document in the Public N&A book to **Yes**.

The screenshot shows the Lotus Notes interface for editing a Server document titled "Stevo/Braintree". The "Security" section is expanded, showing "Security settings" and "SSL settings".

**Basics**

|                              |                                             |
|------------------------------|---------------------------------------------|
| Server name: Stevo/Braintree | Server build number:                        |
| Server title:                | Administrators: Stephen Londergan/Braintree |
| Domain name: Braintree       | Routing tasks: Mail Routing                 |
| Cluster name:                | Server's phone number(s):                   |
| Master address book name:    |                                             |

▶ Server Location Information  
 ▶ Network Configuration  
 ▶ Proxy Configuration  
 ▼ **Security**

**Security settings**

|                                                           |                                                               |
|-----------------------------------------------------------|---------------------------------------------------------------|
| Compare public keys against those stored in Address Book: | <input type="radio"/> Yes <input checked="" type="radio"/> No |
| Allow anonymous Notes connections:                        | <input type="radio"/> Yes <input checked="" type="radio"/> No |
| Allow anonymous HTTP connections:                         | <input checked="" type="radio"/> Yes <input type="radio"/> No |

**SSL settings**

|                               |                                                               |
|-------------------------------|---------------------------------------------------------------|
| Accept SSL site certificates: | <input checked="" type="radio"/> Yes <input type="radio"/> No |
| SSL key file:                 | keyfile.kyr                                                   |

This is the protected text area of the form.

**Figure F.3** Use the Security section of the Server document to configure the Domino server’s use of SSL.

## Creating a Self-Certified Key Ring

Create your own self-Certified Key Ring when you are testing SSL, or when you do not need to rely on a commercial Certificate Authority. This process also creates a Key Ring file to hold the Certificate that is created.

- Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
- Click the **Create Self-Certified Key Ring** button in the **Create Key Ring—Self Certification** section of the Form.
- Enter the desired parts of the Certificate Name. (At the very least, the **Common name, Organization, State, and Country code.**)
- Enter the **Key Ring File Name** and **Password**.

### **Sending a Request to a Commercial Certificate Authority**

Webmasters intending to use “real” SSL on a public Domino server to secure its transactions will request a Certificate from a commercial provider, such as VeriSign. When you create a Certificate Request according to the following steps, you also create a Key Ring file that will hold the Certificate once it is returned from the CA, and another file that contains the actual request. This second file is delivered (e-mail or snail mail) to the CA.

1. Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
2. Click the **Create key ring and certificate request** button in the **Create Key Ring—Commercial Certification** section of the Form.
3. Enter the appropriate components of the Domino server’s name and the **Key Ring File Name** and **Password**.
4. Enter the name of the file that will be delivered to the CA.
5. Deliver the Certificate Request file created in Step #4 to the CA. How you do this depends on the CA; some ask you to e-mail the file, others require more traditional delivery messages (like FedEx), and others request that you upload the file to an FTP server.



#### **NOTE**

If you are using VeriSign as the CA, use the **Create mail message for VeriSign** button to automatically compose and send an Internet e-mail message to VeriSign; the Certificate will be sent to the address listed in the **To:** field.

### Merging a Certificate from a Commercial Certificate Authority

The Certificate Request that you created and sent in the previous section is returned to you (eventually) by the commercial CA; upon receipt, the file must be merged into the Domino server's Key Ring to enable SSL and use the new Certificate.

1. Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
  - Click the **Merge certificate into key ring** button in the Create Key Ring—Commercial Certification section of the Form.
  - Enter the **Name** and **Password** of the Key Ring file that will store the new Certificate.
  - Enter the name of the file containing the new, signed Certificate that was returned to you by the Commercial CA.

### Sending a Certificate Request to an Internal Certificate Authority

If you or another Domino Administrator in your organization has created an SSL Certificate, use the following steps to create a new Key Ring file and request a Certificate from the Internal CA.

1. Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
2. Click the **Create key ring and certificate request** button in the Create Key Ring—Internal Certification section of the Form.
3. Enter the appropriate parts of the Domino server's name.
4. Enter a **File Name** and **Password** for the new Key Ring.
5. Enter the name of the file that will store the Certificate Request.



---

Use the **Send to Certificate Request** button to send the request via e-mail.

---



### Merging a Certificate from an Internal Certificate Authority

The Certificate Request that you created and sent in the previous section is returned to you by the Internal CA; upon receipt, the file must be merged into the Domino server's Key Ring.

1. Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
2. Click the **Merge Signed Certificate into Key Ring** button in the Create Key Ring—Internal Certification section of the Form.
3. Enter the name of the file containing the new, signed Certificate that was returned to you by the Internal CA.
4. Enter the **Name** of the new CA's Certificate file name.
5. Enter the **Name** and **Password** for the Key Ring file name that will store the new Certificate.

### Becoming an Internal Certificate Authority

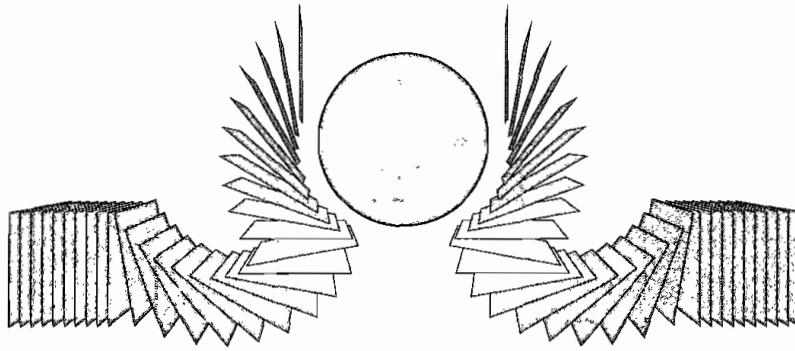
Use the following steps to create a new Certificate that can be used by Domino servers and Web users within your organization.

1. Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
2. Click the **Create Certification Authority Key Ring** button in the Certificate Authority—Create Certificate Authority Key Ring section of the Form.
3. Enter the appropriate parts of the new Certificate's **Name**.
4. Enter the **Name** of the new Key.
5. Enter a **File Name** and **Password** for the new Key Ring.
6. Enter the **Name** of the "Certificate Authority Certificate" file (you use this file when responding to Certificate Requests from other people in your organization/company.)

### Responding to Certificate Requests

As the person who created and owns the Certificate that was created in the previous section, it is your responsibility to respond to Certificate Requests. When you receive a Certificate Request, **Copy** the file to the Domino server's hard disk, and use the following steps to process it.

1. Open the **SSL Administration database** at the server and click the picture of the keys to proceed.
2. Click the **Sign Certificate Request** button in the Certificate Authority—Certify Certificate Request section of the Form.
3. Enter the **Name** of the Certificate Request file (that you received).
4. Enter the **Name** of the new Key.
5. Enter the **Validity Period** (after which the Certificate will no longer function).
6. Enter the **File Name** and **Password** for the Key Ring that will be used to process the request.
7. Enter a name for the file that will contained the signed and processed Certificate request. (This is the file that you will return to the person who requested the Certificate in the first place.)
8. Return the file created in Step #7 to the person who requested the Certificate.



# GLOSSARY

## **Access Control List (ACL)**

A database's ACL is where the users of the database are defined, along with their corresponding levels of access. Domino Access levels are: No Access, Reader, Author, Editor, Designer, and Manager. When a user opens (or tries to open) a database/Web area, the Domino server compares her name against those listed in the ACL, and decides what the person can—or can't—do in the database. Each database has its own Access Control List.

## **Access Log**

An ASCII text file which can be used to log Domino server usage and performance access statistics. Domino creates a new ASCII file log every day at midnight, or when the server is restarted, if it has been configured to do so via the Server document.

## **Agent Log**

A special Notes database that records information when database agents are run.

**Agent**

A collection of commands within a Notes database that are used to automate some sequence of events. For example, an agent could perform a simple task, such as automatically forwarding documents to your assistant while you're on vacation, or register new user accounts on a Domino server.

**Applet**

A small program written in the Java programming language.

**Article**

A message posted in a Usenet newsgroup. Newsgroup readers post articles in a newsgroup and other readers can then post replies to that article.

**Attachments**

A file (such as a Microsoft Word document or Lotus 1-2-3 spreadsheet) that has been stored within a Notes document. Attachments can be included in Domino Web pages.

**Authentication**

The process of establishing a user's identity. Lotus Domino uses the "challenge-and-response" mechanism to authenticate a person, wherein a user is prompted to enter a User Name and Password.

**Bandwidth**

A general networking term used to describe the amount of data that can be transferred across a network, phone, or communications line. (The more you have of this, the better.)

**Browser**

Software run on a person's workstation that can connect to a Web server, download Web pages, and display them. Popular examples of Web browser programs include Netscape's Navigator, Microsoft's Internet Explorer, and the Notes workstation software itself.

**Cache**

The process of storing Web pages on either the workstation or server in an effort to reduce repeated access to a Web site. If a user contacts a site that has been previously retrieved, the contents are retrieved from the cache rather than the actual Web site. The Domino server also caches images so that they do not have to be re-converted each time they are requested by a user.

**CGI error Log**

An ASCII text file stored on the Domino server that is used to trap errors that may occur during the execution of a CGI script.

**CGI scripts**

Programs used by traditional Web servers to automate processes on a Web site. In most cases, Domino eliminates the need for CGI scripts.

**Circuit-level Proxies**

Applications that work "above" the TCP/IP Port or application layer of the protocol to allow clients to pass through a central service on the local LAN to connect to whichever TCP/IP Port the user requests. A SOCKS server can also prevent unauthorized nodes from accessing the Internet. For example, a SOCKS Proxy server could be used to allow local users to connect to Web servers on the Internet, but prevent the local Domino servers from issuing HTTP requests destined for Web servers outside of the local LAN.

**Connection document**

A document in a Notes Name and Address Book that contains information about how the Domino server should connect to the server. Workstations also use Connection documents to schedule connections to a Domino server.

**Cookie**

An ASCII text file created by the Domino server and stored on the user's workstation to record environment variables produced by a Web server.

**DocLink**

An icon that can be pasted into a Notes document which, when double-clicked, jumps to the indicated document. When a Notes document is converted to HTML for display via Domino, DocLinks also function as Links on the Web page.

**Documents**

Each separate entry in a Notes database. When translated to the Web, each Notes document in the database becomes a separate Web page.

**Domain**

A collection of Notes servers.

**Domino Server**

An R4.5-based server from Lotus Development that runs the Domino server software along with the Notes server software and allows multiple Web and Notes client users to view its contents.

**Domino.Action**

A suite of software and database templates that allow you to quickly assemble a Web site powered by Lotus Domino.

**Encryption**

The process of scrambling information to prevent it from being seen by unauthorized users. Domino does not support field or channel encryption, but does use encryption during SSL transactions.

**Error Log**

An ASCII text file that optionally records any problems that may have occurred during client connections. You should check this Log regularly to see if users are encountering problems with your Web site. This Log is configured via the Server document.

**Frequently Asked Questions list (FAQ)**

An Internet convention for a text file that contains commonly-asked questions and their answers. FAQs are typically found in Usenet newsgroups to help readers find their way around. FAQs are now also being found on most Web sites to assist new users, and are included in Domino.Action Web sites.

**Fields**

The building block of a Notes database Form; each field represents a single piece of information to be collected. You can create a variety of different field types. For instance, in a customer order entry form you might find fields such as Customer Name, Product Price, Date Ordered, etc.

**Finger**

A somewhat outdated Internet protocol for establishing a user's identity and log-in status on a host. The Lotus Notes Web browser supports the use of FINGER links.

**Firewall**

A separate computer that sits between a private LAN and the Internet and acts as a protective measure between the two networks. That is, the Firewall is the only computer that is physically connected to the Internet. Typically, a Firewall is configured to filter the information it receives, so that only the appropriate data can be passed between the two networks.

**Folder**

An organizational tool that organizes the documents in a database. A Folder is also similar to View in that a Folder can also be used to navigate from one document to the next.

**Form**

The design element of a Domino application through which documents are displayed.

**File Transfer Protocol (FTP)**

An Internet protocol used to transfer files from one computer to another. FTP also refers to the actual application used to move files using the FTP protocol. The Lotus Notes Web browser supports the use of FTP links.

**Full-Text Index**

A procedure that records words and phrases in all documents in a database so that you can later search for particular documents based on those words.

**Gopher**

A somewhat outdated Internet application that uses a menu system to allow users to search for and download information.

**Graphic Interchange Format (GIF)**

A graphic image file format, which is best at reducing the size of images that contain relatively few colors (such as an icon, or illustration, etc.)

**Group**

A collection of User Names stored in the Domino server's Public N&A Book, and which is used in Access Control Lists to prescribe a level of access to an application.

**Hierarchy**

An organizational scheme used by Usenet Newsgroups. Each newsgroup is organized into separate categories, or hierarchies, each of which has a distinct name and topic focus. For example, the *rec* hierarchy focuses on recreation topics. The newsgroup *rec.sports.tennis* specifically focuses on tennis as a recreational sport.

**Home Page**

A generic term now used universally for the hypertext document users see when they first access the Web or a particular Web site.



**Host Name**

The name of a computer connected to the Internet, which corresponds to its TCP/IP address.

**Hotspots**

Special areas on a Form that can perform some combination sequence of menu commands and formulas. A Hotspot can open another database, document, View, or Web page.

**HyperText Markup Language (HTML)**

The format/language used to write and store Web pages. Domino automatically converts the pages and parts of a Notes database into HTML pages.

**HTML Passthrough**

The process of including HTML tags directly in a Domino application.

**HTML Tags**

The codes used in HTML file to tell a Web browser how the page should be displayed. HTML tags are always enclosed in angle brackets (<>). Note also that attributes such as bold are turned on with a code inside the brackets, and then turned off by the same code, in brackets, but preceded by a slash (/). For example, <BOLD>this is a word</BOLD>.

**HyperText Transfer Protocol (HTTP)**

The Internet protocol used to transfer Web pages via TCP/IP between Web servers and Web browsers.

**HTTP Password**

A Password used to gain access to information on a Domino Web site. A Notes user will have the same level of access and rights when using the applications via Domino as he would when using his Lotus Notes workstation. Domino User Passwords are stored in a field by this name in the Person document in the Public N&A Book.

**HTTP Proxy**

Allows users on a corporate LAN to access the Internet through the Firewall and uses HTTP in all communication between the user and the Firewall server. The HTTP Proxy limits the Internet protocols required within the local LAN, and greatly simplifies the configuration of the Proxy and Firewall since all communication between the local users and the Proxy is carried over HTTP. As a result, the Firewall server can then be configured to exclude all of the other Internet protocols, such as FTP, WAIS, and so on.

**HTTPS**

The beginning of a URL that connects to a web site secured by SSL. Both the Domino server and Notes Web browser support the use of this URL.

**Hypertext**

Rich text within an online document that typically contains links, images, and other multimedia-based content.

**Image Map**

An image that Web users click in lieu of Hypertext links. In Domino, database Navigators are converted into Image maps.

**Internet**

The worldwide collection of computers that are connected via the TCP/IP network protocol.

**Internet Service Provider (ISP)**

A company which sells connections to the Internet.

**InterNotes server**

An outdated term for a Domino server.

**InterNotes News Gateway**

A news software program that allows a Domino server to receive and post Usenet articles.

**Intranet**

A collection of computers within a given company or organization that contain Web pages, but which are not accessible by people outside of the organization or company. In an Intranet, Web clients connect directly to the Web server via the company's LAN. Intranets let companies use Web technology without the security headaches of opening their LAN to the world. In most cases, you can substitute the term *Intranet* for *Internet* as appropriate.

**IP address**

The numeric TCP/IP address of a computer. See *Host Name*.

**Java**

A programming language that is used to create small, cross-platform programs (or *applets*) that can be included in Web pages. An applet is downloaded to a workstation's memory and executed when a Web browser encounters a "Java-enabled" Web page. Most Web browsers (including Notes) can interpret, compile, and run Java applets.

**JPEG (Joint Photographic Experts Group)**

Also JPG. A graphic file format quickly gaining speed as the standard file format for graphics on the Web, which is extremely efficient at reducing the size of image files, particularly those that are based on photographs. Domino servers convert images within a database to either JPEG or GIF files.

**Launch Property**

The information that is displayed when a database is opened. Choices include: Restore as last viewed by user, Open "About Database" document, Open Designated Navigator, Open Designated Navigator in its own window, Launch 1st attachment in "About Database", and Launch 1st DocLink in "About Database".

**Link**

Part of a Web page that, when clicked by a user, opens a different Web page. Links are the cornerstone of the Web, as they provide users with an easy way to navigate between different documents, databases, applications, and entire Web sites.

**Location document**

A document in the Personal N&A Book that contains information regarding the location of a workstation. The Location document is also used to configure the manner in which Web pages are retrieved on that workstation, at that location.

**Mirrored Site**

Web server that is an exact duplicate of another, used for load balancing. Domino can use replication to create mirror sites.

**Multi-Database Searching**

The process of creating a Full-Text Index that spans multiple databases. When transferred to the Web, this becomes a valuable tool, as users will be able to search multiple areas of a Web site from a single point of access.

**Multihoming**

The process of using a single Domino server as the location of more than one Web site. In other words, one computer that is both *www.site1.com* and *www.site2.com*.

**Navigator**

A design element in a database which is converted to an Image Map by Domino.

**Netscape Plug-ins**

Small programs that are used in conjunction with a Web page's content to deliver customized content along with a Web page to the user. Plug-ins are similar to Java applets in that they provide additional functionality to the user. Unlike Java applets, however, Plug-ins must be installed on the user's workstation before the content is loaded. The Notes Web browser supports the use of Plug-ins.

**Newsgroups**

Collections of discussion groups that reside on various Usenet servers on the Internet. Newsgroups resemble electronic bulletin boards where users around the world read, post, and reply to articles posted by other newsgroup readers.

**News database**

The Notes database that Notes users open to read and post Usenet articles.

**NOTES.INI**

The ASCII file stored on every Notes workstation and Domino server that contains its configuration information.

**Notes client**

The software used on a person's workstation to open Notes databases and, possibly, retrieve Web pages.

**Notes Public Networks (NPN)**

Wide Area Networks that have been optimized exclusively for Notes and are provided by telecom carriers such as AT&T, Compuserve, and IBM. By using a NPN, you can outsource both the hardware and the necessary support required to run a large, worldwide WAN.

**Notes server**

An outdated term for a Domino server.

**PageMinder**

A special agent in the Personal Web Navigator database that notifies a user when an identified page on the Web has changed.

**Packet**

A package of information that has been broken down by TCP in order to be transferred by IP. For example, if you post an article to a newsgroup, TCP breaks down your article into small packets of information that IP then transfer over the Internet.

**Personal Web Navigator**

A database on a Notes workstation which stores the Web pages that have been previously retrieved by the user. This database stores pages so that subsequent requests for the same page can be served up from this database instead of from the originating Web server. Users can the re-read the Web pages they've visited, even when the workstation is not connected to the Internet.

**POP3**

A simple mail protocol supported by Domino servers.

**PPP**

Point-to-Point Protocol is the dial-up connection protocol that enables a computer to use a modem to connect to other modems on the Internet.

**Proxy server**

A special computer that provides access to information (such as URLs or network protocols) through a Firewall server, and which typically masks the identity of the person or server requesting the page. Both Domino servers and the Notes workstation can retrieve Web pages through a Proxy server.

**Proxy**

An application that runs on the Firewall server and presents a "public face" to the Internet. When a user behind the Firewall/Proxy requests a Web page, the request is passed to the Proxy, which in turn passes the request to the Web server on the Internet. The Proxy receives the Web page back from the Web server, and forwards the page to the user who requested it in the first place.

**Public N&A Book**

A special database on the Domino server that contains virtually all of the information regarding the server's Configuration.

**Public Web Navigator**

A database on the Domino server which stores the Web pages retrieved by the Domino server. Subsequent requests for such a page are then served by the Public Web Navigator rather than having to reload the actual Web site.

**Refer Log**

An ASCII file on the Domino server used to which link a user clicked to open a page on the Domino server.

**Refresh**

The process of updating the contents of a previously-viewed Web page or site.

**Replica**

A copy of a Notes database.

**Replication**

The process of synchronizing the documents and design elements between two replica copies of a database; either between two servers or between a server and a workstation.

**Sections**

Areas of a Form that are used to control access so that only certain people can access fields in a section. Domino supports sections.

**Secured Socket Layer (SSL)**

A protocol that encrypts the network traffic between the Domino server and the Web browser. Since the WAN or LAN packets between the server and client are encrypted, users can be confident that transmitting information such as home addresses, telephone numbers, and credit card numbers can not be compromised. Adding an extra "s" to a Domino URL (such as <https://www.steve-o.com/>) automatically invokes the SSL encryption ports and algorithms.

**Simple Mail Transfer Protocol (SMTP)**

A member of the TCP/IP family of protocols and is the standard used for formatting and sending e-mail via the Internet.

**Source server**

The Notes server that will be initiating an Internet connection, as defined in a Connection document.

**Spool database**

The database used to temporarily store Usenet articles on their way to and from the user's workstation and Usenet database.

**Subscribe**

The process of joining a newsgroup. After subscribing to a newsgroup, new articles and responses to articles are sent to you. Note that in some instances, your network administrator or company Webmaster may have to provide you with the necessary access.

**TCP/IP**

The Transmission Control Protocol (TCP) is one of the main protocols used on the Internet to break down data into individual packets of information so that it can be transferred using the Internet Protocol (IP).

**Universal (or Uniform) Resource Locator (URL)**

The address of a particular Web server, Web page, or other Web resource. For instance, *<http://www.microsoft.com/games/>* or *<https://domino.lotus.com.whats-new.html>*.

**Usenet**

A worldwide collection of computers that store and exchange news among other computers. The Internet connects most of the Usenet computers together.

**View**

A design element of a database that summarizes the documents in the database.

**View Links**

Similar to DocLinks, except that they open a particular database's View, rather than an individual document.

**WebAhead**

An agent in the Personal Web Navigator database that retrieves Web pages for offline reading.

**Webmaster**

the person responsible for the care, feeding, and general maintenance of the Domino server. Also known as the Domino Master.



**Web page**

A multimedia document that is stored on a Web server and viewed by people using a Web browser program. On a Domino server, any document, View, Form, or Navigator has the potential to become a Web page.

**Web publisher**

Generic term for a tool or software package that converts documents to HTML so that they can be viewed on the World Wide Web.

**Web server**

A computer that holds Web pages, and distributes them to Web browsers. Web server software is manufactured by many different companies, including Lotus.

**Web site**

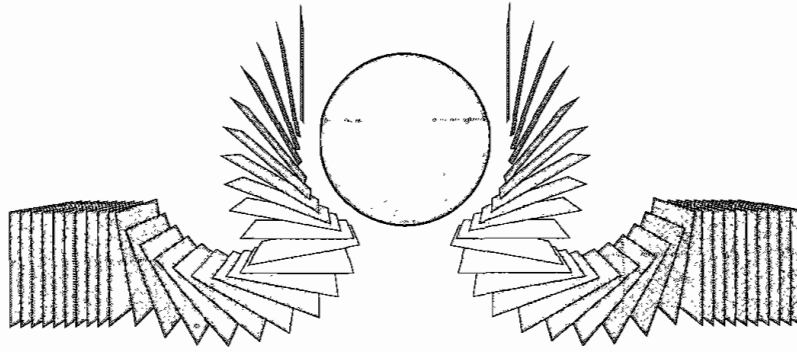
The set of pages that comprise a person or organization's presence on the World Wide Web.

**World Wide Web (WWW)**

The collection of computers (or servers) on the Internet that store and deliver Web pages. Referred to (sarcastically) as the "World Wide Wait," sometimes.

**Workflow**

A programmed application that pushes or pulls information along a pre-defined process, usually through the use of macros or agents. Lotus Notes is widely known for its Workflow capabilities, and this functionality extends to Domino applications on the Web.



# INDEX

\$\$Fields and \$\$Forms, 550-552

@Commands, 558-559

@Functions, 557-558

@URLGetHeader, 332

@URLHistory, 333

@URLOpen, 322-331

arguments, 329

## A

About document

creating, 133

Access Control List, 50-53

assigning access levels, 51

defined, 581

listing Web users, 227

within an application, 229

Access Log

defined, 581

Agent , 62-63

creating, 62

defined, 582

Agent Log

defined, 581

Applet

defined, 582

Java, 80

Application design

standards, 71-80

- Article, 354
  - defined, 582
- Attachments, 174-175
  - defined, 582
- Authentication, 217-225
  - access control, 226
  - Anonymous users, 222
  - defined, 582
  - Non-Notes Web users, 221
  - workstation users, 219
- Author access, 53
- Authors, 59
- B**
- Background, 175
  - creating, 175
  - limiting use of, 78
- Bandwidth
  - defined, 582
- Browser
  - defined, 582
  - using an external browser, 254
- C**
- Cache
  - defined, 583
- Cache directory, 95
- Certificate Authority
  - described, 572
  - merging a certificate, 578
  - responding to certificate requests, 580
  - sending a request, 577
- CGI-BIN directory, 89
- CGI error Log
  - defined, 583
- CGI scripts
  - defined, 583
- CGI variables, 153-155
  - capturing, 153
  - using cookies, 154
- Character sets, 98-99
- Circuit-level proxies
  - defined, 583
- Computed fields, 59-60
- Connection document
  - defined, 583
- Conversion, 139-145
  - actions, 144
  - agents, 145
  - field, 142
  - form, 141
  - navigator, 144
  - scripts, 145

- text, 139
- view, 142
- Cookie
  - defined, 583
  - using, 154
- Custom search agents, 335
- D**
- Database
  - defined, 584
  - sample applications, 26-30
- Database Link
  - defined, 584
  - described, 61
- Depositor access, 52
- Design Element Cache, 104
  - defined, 584
- Designer access, 53
- Destination server
  - defined, 584
- Directory links, 88
- Disk Cache, 103
  - defined, 584
- DocLink
  - defined, 585
  - described, 61
- Documents
  - creating, 117
  - defined, 585
  - editing, 118
  - opening, 117
- Domain
  - defined, 585
- Domain Name Server (DNS)
  - defined, 584
- Domino, 4-11
  - sample applications, 26-30
  - site planning, 67-71
- Domino.Action
  - Access control, 495
  - approval agents, 507
  - assigning a Home page, 506
  - assigning roles, 500
  - assigning users, 459
  - configuring, 435
  - contributor menu, 519
  - creating a full-text index, 505
  - creating a Library database, 457
  - creating a SiteCreator database, 457
  - defined, 585
  - defining approval process, 514
  - described, 431-433
  - designing, 436, 442-453
  - editing Main pages, 503
  - generating the site, 487-489

- registration, 462
  - registration, 508
  - Domino Links, 136-139
    - creating, 136
    - with formulas, 138
  - Domino News Configuration Database, 383
    - assigning user access, 391-393
    - creating a full-text index, 394
    - managing connections, 385
    - managing database size, 393
    - managing News databases, 387
    - managing News gateways, 388
    - managing News servers, 388
    - managing subscriptions, 384
    - subscribing, 389
    - unsubscribing, 390
    - viewing the connection log, 386
  - Domino Server
    - accessing from the Internet, 415
    - application-level security, 418
    - checking performance, 102
    - configuring, 90-100
    - defined, 585
    - described, 40
    - enabling SSL transactions, 576
    - hardware requirements, 85-86
    - installing, 86
    - logging, 97-98
    - network-level security, 423
    - optimizing, 104-107
    - overview, 3-11
    - retrieving Web pages, 243
    - software components, 87
    - using modems, 427
    - using multiple LAN cards, 427
  - Domino views
    - and ReaderFields, 182
    - assigning access, 181
    - customizing, 186
    - opening, 180
- E**
- Editable fields, 59
  - Editor access, 53
  - Electronic commerce, 29
  - Encryption
    - defined, 585
  - Error Log
    - defined, 585
- F**
- Frequently Asked Questions list (FAQ)
    - defined, 586

Fields, 57, 59

    creating, 60

    defined, 586

    HTML, 158

Finger

    defined, 586

Firewall , 423

    defined, 586

    packet-filter, 424

    proxy, 423

Folder

    defined, 586

Form, 57-58

    creating, 58, 161

    customizing response form, 171

    defined, 586

File Transfer Protocol (FTP)

    defined, 587

Full-Text Index, 201-204

    creating, 202

    defined, 587

    updating, 204

## G

Garbage collection, 95

GIF directory, 89

Gopher

    defined, 587

Graphic Interchange Format (GIF)

    defined, 587

Group

    defined, 587

## H

Hierarchy, 354

    defined, 587

Home Page

    assigning, 148-150

    creating, 130

    defined, 587

    HTML file, 150

    URL redirection, 151

Host Name

    defined, 588

Hotspots , 57, 61

    defined, 588

    Link vs Action, 75

HTML directory, 89

HTML header, 159

HTML Passthrough

    defined, 588

    described, 155

    Direct passthrough, 156

    in a field, 163

    with images, 73

- HTML Tags
  - creating, 39
  - defined, 588
  - described, 38
- HTTP Password, 45
  - assigning, 124
  - defined, 588
- HTTP Proxy, 425
  - defined, 589
- HTTPS
  - defined, 589
- Hypertext
  - defined, 589
- HyperText Markup Language (HTML)
  - defined, 35, 588
  - described, 37
- HyperText Transfer Protocol (HTTP)
  - defined, 35, 588
- I**
- Image conversion, 75
- Image Map
  - defined, 589
- Input validation, 165
- Internet, 19
  - defined, 34, 589
- InterNotes server
  - defined, 589
- InterNotes News Gateway, 17, 353, 358-364
  - choosing a server, 531
  - console quick reference, 535
  - creating a News Connection document, 372
  - creating a News Gateway document, 370
  - creating a News Server document, 371
  - defined, 589
  - described, 359-364
  - downloading Newsgroups, 375
  - hardware requirements, 366
  - installing, 366-368
  - loading the INNEWS server task, 369
  - News configuration database, 358
  - running for the first time, 368
  - sample newsgroups, 534-535
  - scheduling News transfers, 373-374
  - Spool database, 358
- Internet Service Provider, 404
  - choosing, 536
  - connecting Notes to the Internet, 404
  - defined, 589

Intranet  
defined, 34, 590

IP address  
defined, 590

## J

Java, 341-345  
controlling execution, 344  
defined, 590  
running applets, 342  
JPEG (Joint Photographic Experts Group)  
defined, 590

## K

Key ring, 572  
Keywords, 59

## L

Launch property, 111  
assigning, 133  
defined, 590  
site launch, 134  
Link  
creating, 75, 549  
defined, 34, 590

Location document  
defined, 591  
managing, 346

## M

Mail, 16, 121-127  
configuring, 123-127  
reading with a browser, 122  
replicating, 126  
Manager access, 53  
Mapping, 94  
Message encryption, 571  
Message validation, 572  
Microsoft Internet Explorer, 15  
Mirrored Site  
defined, 591  
Multi-Database Searching  
configuring, 213  
creating a search site, 214  
creating search scope documents, 214  
defined, 591  
indexing, 215  
Multihoming  
defined, 591



**N**

Names, 59

Named style, 157

Navigator, 63-65

- converting to image maps, 176

- creating, 65

- defined, 591

- described, 63

- in a Web application, 177

- persistent, 76

Netscape Navigator, 15

Netscape Plugins, 345-346

- defined, 591

Network protocols, 244

Newsgroups, 354

- defined, 591

- downloading, 375

News database, 359

- configuring News Gateway, 378

- creating, 376

- defined, 592

- managing, 387

No access level, defined, 52

NOTES.INI

- defined, 592

Notes, connecting to the Internet,  
404-416

- enabling TCP/IP, 405-407

Notes client

- defined, 592

Notes Public Networks (NPN), 404

- connecting Notes to the Internet,  
407

- defined, 592

Notes server

- connecting through the Internet,  
412

- creating a connection document,  
413

- defined, 592

Notes Web Browser

- configuring, 260

Notes workstation

- configuring browser, 263

- connecting to a Server through  
the Internet, 408

- creating a connection document,  
410

- creating a location document, 409

- retrieving Web pages, 250

- with a proxy server, 252

**O**

Offline browsing, 29

**P****Packet**

- defined, 592

**PageMinder, 254, 293-296**

- defined, 592

- configuring, 267, 294

**Page titles, 78****Personal Web Navigator, 241, 253-254**

- configuring, 264, 283-287

- customizing, 333

- defined, 592

- described, 279-280

- enabling PageMinder Agent, 294

- enabling WebAhead agent, 299

- managing size, 269

- opening a Web page, 281

- startup options, 265

- using, 287-293

**POP3, 16**

- defined, 593

**PPP**

- defined, 593

**Proxy server, 245**

- defined, 593

- configuring, 348

**Proxy, 423**

- circuit-level, 426

- defined, 593

- SOCKS, 426

**Public N&A Book**

- defined, 593

**Public Web Navigator, 248, 301-303**

- averaging agent, 318

- configuring server document

- configuring, 270, 308

- customizing, 333

- defined, 593

- described, 248-250

- opening a Web page, 303

- purging documents, 315

- refreshing, 316

- registering users, 306

- using, 311-315

- workstation location documents, 305

**R****Reader access, 53****Readers, 59****ReaderFields, 182****Refer Log**

- defined, 593

- Refresh
    - defined, 594
  - Registration Database
    - adding to a Web site, 235
    - using, 232
  - Replica
    - defined, 594
  - Replication
    - defined, 594
  - Revision tags, 79
  - Rich text, 59
- S**
- Search, 119-121
    - Refine search, 120
    - Search button, 119
  - Search form, 204
    - copying from Search Site
    - customizing search results, 209
    - customizing, 206
  - Sections, 57
    - defined, 594
  - Secured Socket Layer (SSL), 9, 425, 571-580
    - defined, 594
    - enabling, 576
    - SSL certificate name, 574
    - SSL key ring, 575
    - using in a Domino server, 573
  - Security, 44-45
    - for Domino servers, 418-427
    - workstation vs browser, 237
  - Security profile, 495
  - Selection criteria, 54
  - Server authentication, 572
  - Server Console, 100-102
  - Simple Mail Transfer Protocol (SMTP)
    - defined, 594
  - SMTP Message Transfer Agent, 106
  - Source server
    - defined, 594
  - Spool database, 358
    - defined, 594
  - Static text, 57
  - Subscribe, 379
    - defined, 595
  - Submit button
    - creating, 169
    - customizing, 167
- T**
- TCP/IP
    - defined, 595
    - enabling on a server, 405
    - enabling on a workstation, 406
  - Threads, 106
  - Timeouts, 94-95

**U**

Universal Resource Locator (URL),  
43-44

defined, 34, 595

Domino URL command syntax, 539

URL Link, 61

defined, 595

USENET, 353-355

defined, 595

**V**

View, 41-42, 53-57

creating, 55

opening, 116

Selection criteria, 54

View Links

defined, 595

described, 61

View templates

creating, 194

default, 199

described, 189

designing, 195

Virtual servers, 561-568

Domino configuration database,  
562

Mapping directories, 564

mapping URLs, 565

redirecting URLs, 567

Virtual server configuration  
document, 563

**W**

WebAhead, 254, 297-299

configuring, 266

defined, 595

Web browser, 11

defined, 34

Weblicator

described, 569-570

Webmaster

defined, 595

Web page

defined, 34, 596

Web publisher

defined, 596

Web server

defined, 34, 596

described, 35-36

Web site

defined, 596

World Wide Web, 19

defined, 34, 596

Workflow

defined, 596

## COMPANION CD-ROM CONTENTS

All of the files on the CD are Notes databases and should be copied into the Domino server's data directory to be used (usually c:).

1. The DomLog folder contains a customized database template that can be used to create a modified Domino server log. This template has several new views that sort and organize the documents in the Domino log database, as well as a few relatively minor cosmetic enhancements. Copy the files in this folder on the CD to the Domino server's data directory to use them. (Before copying these files from the CD, you should back up your existing domlog.ntf database templates.)
2. The News folder contains two database templates, which can be used to create customized News Configuration and News databases, respectively. Copy the files in this folder on the CD to the Domino server's data directory to use them. (Before copying these files from the CD, you should back up your existing inconfig.ntf and news.ntf database templates.)
3. The Basic, Contemporary, and Corporate folders each contain an entire Web site, created with Domino.Action. Since creating a Web site with Domino.Action can take more than an hour, we have included samples with each "look" on the CD so that you can browse each type of site to evaluate its respective appearance. Each of these folders must be copied to the Domino server's data directory. In other words, copy the entire Basic, Corporate, and Contemporary folders into the Domino server's data directory; you must use the same folder names for Domino to display the embedded graphics properly.

The databases contained in these folders have the Anonymous level of access set to Manager in the interest of simplicity. Consequently, you should *not* put these databases into production; they are intended for reference use only.

For your reference, the files on the CD are arranged as follows:

### `\notes`

#### `\data`

#### `\DomLog`

- `\domlog.nsf` A sample Domino server's log, based on the customized log template
- `\domlog.ntf` The customized Domino Server Log template

### `\News`

- `\inconfig.ntf` A customized template for creating a News Gateway Configuration database
- `\news.ntf` A customized template for creating a news Gateway News Database

## Companion CD-ROM Contents (continued)

\Basic                      A sample Web site, using the Domino.Action "Basic" look

- \About.nsf
- \CorpPoli.nsf
- \Discuss.nsf
- \DocLibra.nsf
- \Fags.nsf
- \Feedback.nsf
- \HomePage.nsf
- \JobPosti.nsf
- \Products.nsf
- \Registra.nsf
- \WhitePap.nsf

\Contemp                    A sample Web Site, using the Domino.Action "Contemporary" look

- \About.nsf
- \CorpPoli.nsf
- \Discuss.nsf
- \DocLibra.nsf
- \Fags.nsf
- \Feedback.nsf
- \HomePage.nsf
- \JobPosti.nsf
- \Products.nsf
- \Registra.nsf
- \WhitePap.nsf

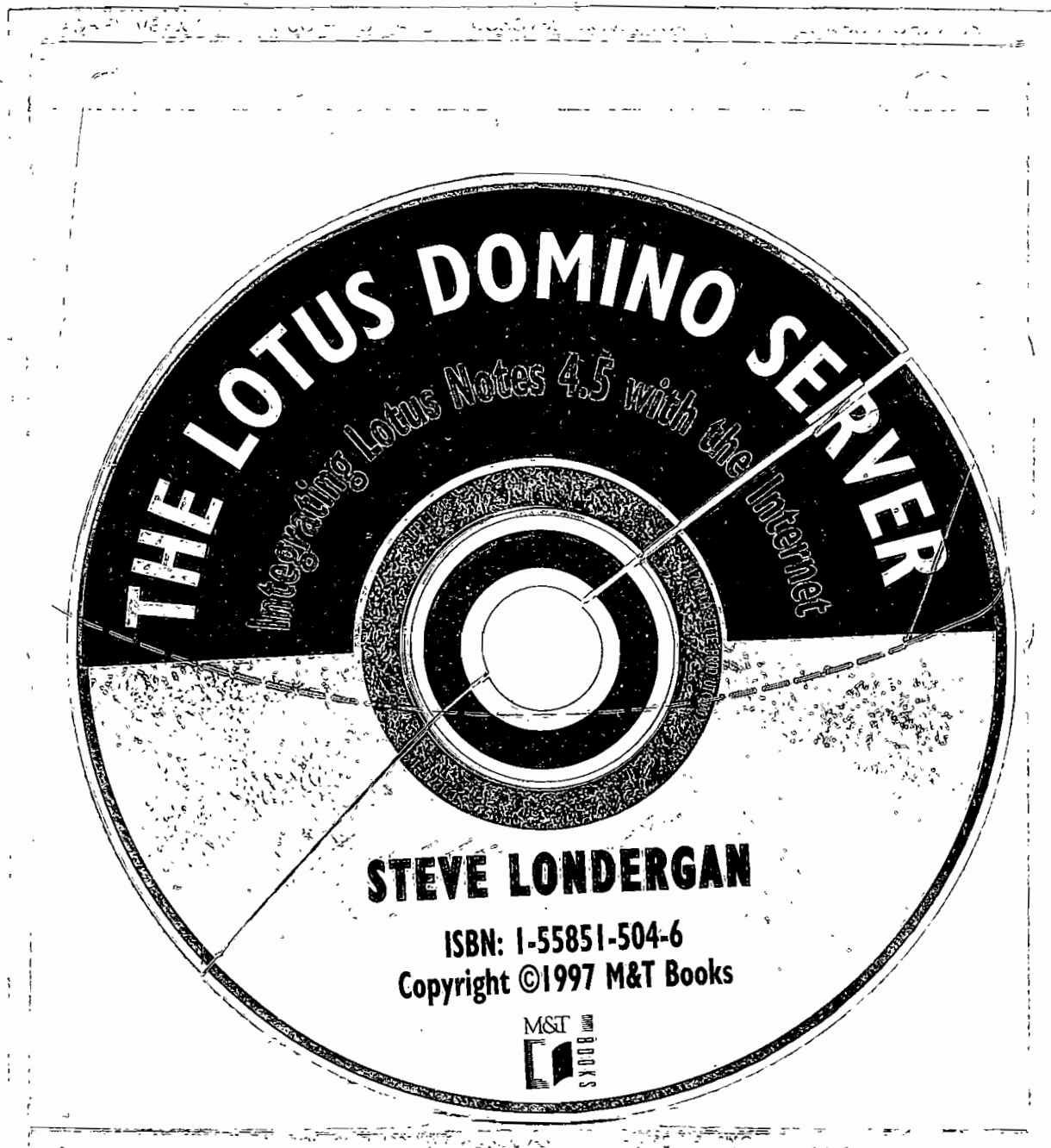
\Corporat                   A sample Web Site, using the Domino.Action "Corporate" look

- \About.nsf
- \CorpPoli.nsf
- \Discuss.nsf
- \DocLibra.nsf
- \Fags.nsf
- \Feedback.nsf
- \HomePage.nsf
- \JobPosti.nsf
- \Products.nsf
- \Registra.nsf
- \WhitePap.nsf



Depending on the type of CD-ROM drive you have, you may have to use your operating system to unmark the files as read-only after copying them to the Domino server's data directory.

---



**T**he Lotus Domino Server packs practical and timely advice for integrating the security, workflow, and replication technologies of Lotus Domino with the World Wide Web. The book explains how to plan, create, and publish applications on the Web with Domino, Lotus' new HTTP server for Notes which transforms the Notes Release 4.5 server into an Internet/Intranet application platform. In addition, this reference focuses on Domino and Web integration, including using the Internet to connect Domino servers, learning how to manage and deploy Lotus's new Weblicator replication technology, and retrieving pages from the Web for Notes users using Notes Web Browser. The configuration and use of Domino.Action, Lotus' groundbreaking tool for building and deploying customized and fully interactive Web sites, is especially emphasized.

The book explores site planning and deployment, server configuration issues, security considerations, and how to customize the Web Navigator Databases. Other advanced topics include building Intranet-based Web sites and integrating Notes with Internet news. The CD includes sample Domino applications and customized database templates for quick Web-based application development and server management. For technical managers and administrators responsible for the planning, deployment, and management of Domino/Web integration, this book is essential reading.

**STEVE LONDERGAN** (Braintree, MA) is an industry-recognized expert in Lotus Notes and has been teaching people how to use, plan, and manage Notes applications and servers since 1990. Until recently he was a senior manager at Lotus, and now manages technical support for Shiva Corporation.

- Details how to plan a site using Lotus Domino
- Explains how to install, configure, and manage a Web site powered by Domino
- Discusses using Notes workstations to browse the Web, as well as the use and management of the Notes Web Navigator database
- Teaches how to build and customize Web sites quickly with Domino.Action
- Provides in-depth details about planning and deploying Internet and Intranet applications
- Offers real-world case studies about integrating Domino with the Internet and Lotus Notes 4.5
- Essential reading for Domino Webmasters, administrators and managers
- Covers Lotus Domino 4.5

Level

Intermediate/Advanced

Lotus Notes/Internet

All Platforms



M&T Books, A Division of MIS:Press, Inc.  
A Subsidiary of Henry Holt and Company, Inc.  
115 West 18th Street  
New York, New York 10011  
<http://www.mandt.com>



Cover design by Gary Szczecina

US \$39.95  
CAN \$55.95

ISBN 1-55851-504-6



9 781558 515048

90000

