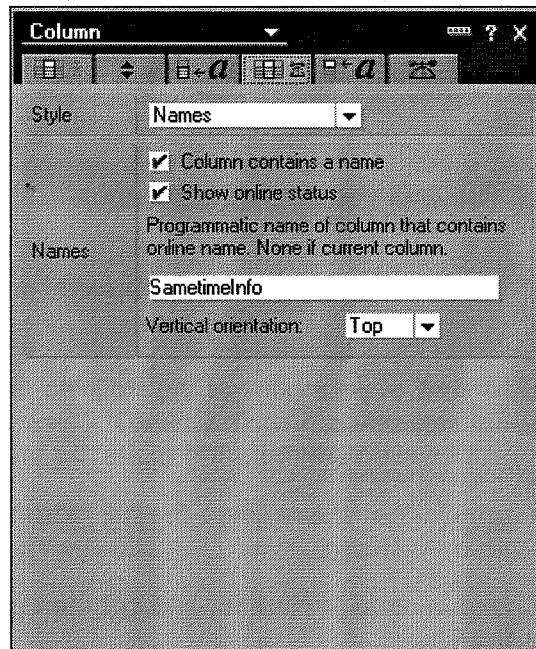


## Names and properties

The following figure shows the instant messaging-related column properties.



**Figure 11-1: Column properties related to instant messaging**



## Optional Practice Activity 11-5: Enable a Column for Instant Messaging

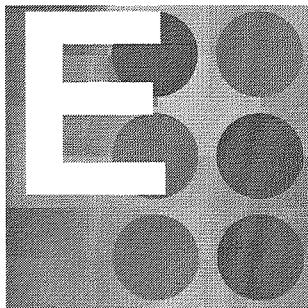


**Scenario:** In an effort to improve interoffice communications, the Worldwide Corporation would like to have users' on-line statuses displayed within the Policies and Procedures database. Follow the steps below to modify the database.



**Note:** The guided solution to this activity can be found in Appendix A, Solutions to Practice Activities.

1. In the Q+A view of your copy of the Policies and Procedures database, create a column called "Who's Online?" Place the new column to the right of the Question column, and make the width at least 18 characters.
2. Enter a formula that abbreviates the full hierarchical name of the contact person into the column.
3. Set the column style to Names, and set the options for the column to show online status.
4. Test your work in the Lotus Notes client.



## Topic E: Create Views for the Web

After completing this topic, you should be able to:

- ✓ List considerations for displaying view in a Web browser.
- ✓ Identify the characteristics of embedded views.
- ✓ Describe view display options, such as the view applet, HTML, and view Display properties.

### View Display Considerations

The default appearance of a view is functional, but does not have the same graphical appeal that is expected by Web users. Views should be explicitly designed with Web display in mind. The first step in improving the appearance of the view on a Web page is to embed it in a page.

### Embedded Views

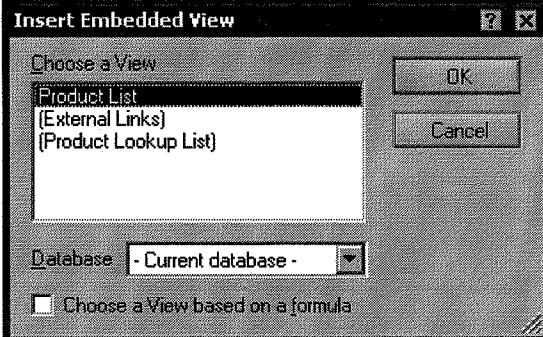
Embedded elements are objects and controls that can be embedded on a page, form, subform, or document. Elements that can be embedded include:

- Views
- Folders
- Outlines
- Navigators
- Date pickers
- Instant Messaging Contact Lists

When you embed a view on a form, subform, page, or document, you can control the size and appearance of a view, especially on the Web. Embedding a view lets you combine views with other form and page elements (such as styled text) and graphics, including a background image file. The generic navigation graphics and links do not appear so that you can replace them with your own.

### Embedding views in forms and pages

Whether you use a page or a form, use the same process to embed a view. Follow these steps to embed a view in a form.

Task	Procedure
1	<p>Choose <b>Create</b>→<b>Embedded Element</b>→<b>View</b>.</p> 
2	<p>Select a view from the list in the current database or from any other database on the server.</p> <p><b>Note:</b> You can also choose the view at run-time via a formula.</p>

### Displaying an embedded view on a form

The following figure shows an embedded view on a form with company identification added.

Health Systems International Product List			
	SKU	Product Name	Price
<b>Biking</b>			
	exer0001	Recumbant Bicycle	\$1000
	exer0002	Treadmill	\$1500
<b>Cross-Training</b>			
	exer0001	Recumbant Bicycle	\$1000
	exer0002	Treadmill	\$1500
	wght0003	Whey Protein	\$25
<b>Hiking</b>			

**Figure 11-2:** An embedded view on a form

### View Display Options

There are three ways to control the display of embedded views on the Web:

- Using a Java applet (the view applet)
- Using HTML
- Using the view's Display property

## Displaying an Embedded View using the View Applet

With the Java view applet, Web users can:

- Resize columns with sliding panes
- Collapse and expand views without the browser regenerating the page
- Select multiple documents
- Scroll vertically to view additional documents in the view
- Press F9 to refresh the view
- Press Delete to mark documents for deletion from the database

Displaying the embedded view using the view applet has advantages and disadvantages. The following table lists some of each.

Advantages	Disadvantages
Select multiple items from the list	Slow to download
Can scroll through large lists of items	May not execute identically on all browsers
Data downloaded on demand	Firewall may prohibit applet download

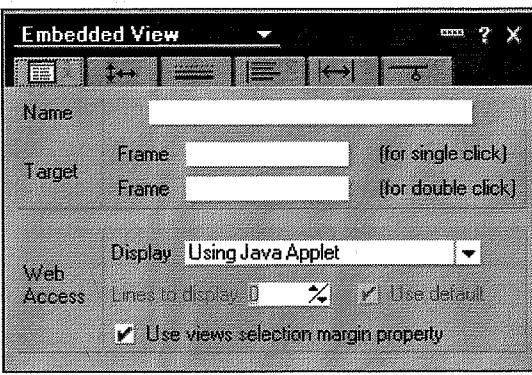
The following figure shows a view applet displayed in a Web browser.

Health Systems International Product List		
SKU	Product Name	Price
<b>Biking</b>		
exer0001	Recumbant Bicycle	\$1000
exer0002	Treadmill	\$1500
<b>Cross-Training</b>		
exer0001	Recumbant Bicycle	\$1000
exer0002	Treadmill	\$1500

Figure 11-3: A view applet displayed in a Web browser

## Displaying an embedded view as a view applet

Follow the steps below to display an embedded view as a view applet.

Task	Procedure
1	Embed the view in a page or form.
2	Open the <b>Embedded View Properties</b> box and select the <b>Info</b> tab. <b>Result:</b> The following graphic shows the <b>Info</b> tab in the <b>Embedded View Properties</b> box. 
3	Select <b>Using Java Applet</b> in the Web Access Display field.

## Displaying an Embedded View using HTML

Displaying an embedded view using HTML is more universal than the view applet because any Web browser can display it. It is not flexible in terms of its capabilities, but it does offer some flexibility in appearance.

The following figure shows a view displayed using HTML.

Health Systems International Product List		
SKU	Product Name	Price
▼ Biking		
	<u>exer0001</u> Recumbant Bicycle	\$1000
	<u>exer0002</u> Treadmill	\$1500

**Figure 11-4:** A view displayed using HTML

## Embedded view properties

The following figure of the Embedded View Properties box shows the display options for an embedded view.

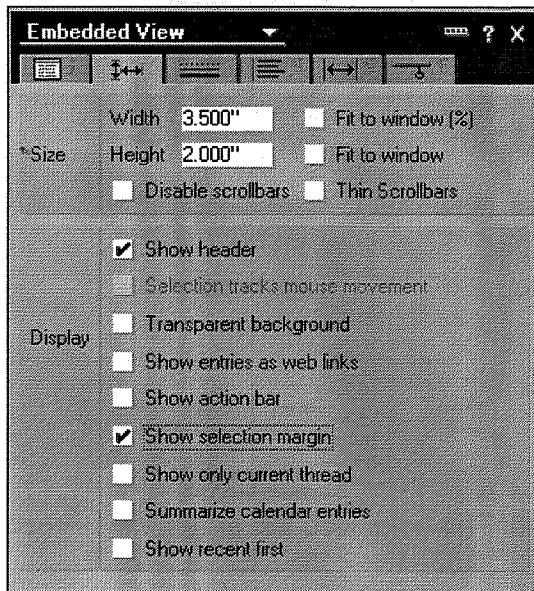


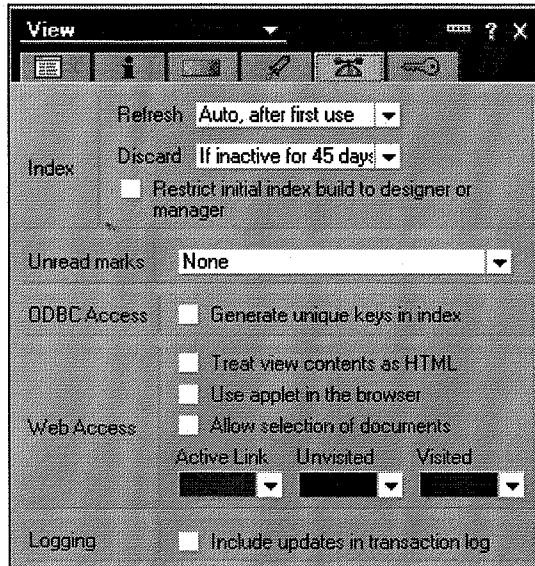
Figure 11-5: Display options for an embedded view

The width and height do not affect the display of an HTML view. Many of the other display options, such as **Show selection margin**, will show up in the resulting HTML view.

## Using View Properties to Control the Display of Embedded Views

You can determine how a browser displays a view using the View Properties box. If the view is embedded in more than one form or view, you only need to set the properties once, rather than setting them in form or page where the view is embedded.

The following figure shows the Advanced tab of the View Properties box.



**Figure 11-6: The Advanced tab of the View Properties box**

### Exploring View Display in the Web Browser

---

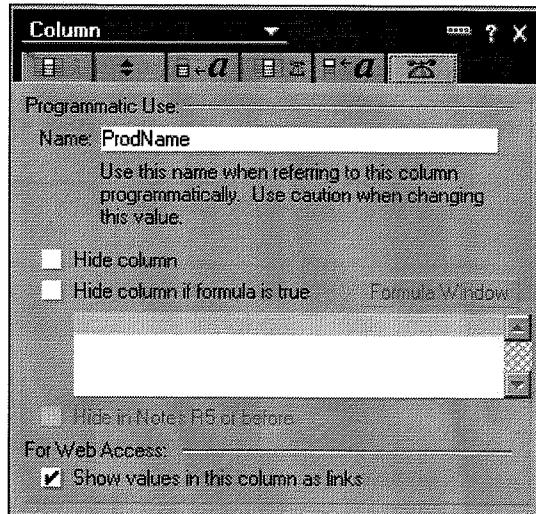
Besides determining how to display your view, you also need to determine other view design effects. Some questions to consider when displaying views in a Web browser are:

- Which column will be used as the link?
- Is there more data than will fit on the screen?
- Will data columns be too close to each other?
- Are graphics needed in view columns?

Some of the more useful properties of views are functional when the view is displayed in a Web browser, including action buttons and **Click on column header to sort**.

## Link Columns

An important design aspect of a view is to select a link column. This is the column of the view that behaves as a link when displayed in a Web browser. By default, the first column is the link column. The following figure shows the **Show values in this column as links** property.



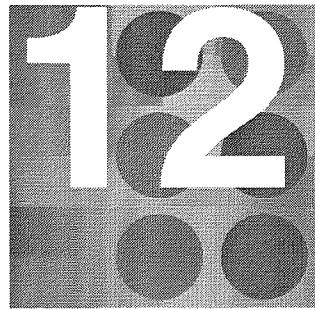
**Figure 11-7: The Show values in this column as links property**



## Activity 11-6: Embed a view on a form

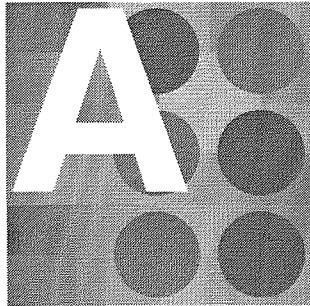
Follow these steps to customize the Products view and embed it in a Lotus Domino form.

Step	Action
1	In Lotus Domino Designer, open the <b>Products</b> view in your copy of the <b>Practice</b> database.
2	Preview the view in your Web browser and note the difficulty in reading it.
3	Open the <b>Product View Display</b> form in the Lotus Domino Designer client.
4	<p>Create an embedded view in the first row of the three-row table located on the right side of the form.</p> <ol style="list-style-type: none"><li>Click the table cell.</li><li>Select <b>Create</b>→<b>Embedded Element</b>→<b>View</b>.</li><li>Select <b>Products</b> from the <b>Insert Embedded View</b> dialog box. Click <b>OK</b>.</li><li>Double-click the embedded view, and then right-click and choose <b>Embedded view</b> to display the Properties box.</li><li>Display the embedded view <b>Using HTML</b>.</li><li>Set the view width to 5 inches, and the height to 3 inches.</li></ol>
5	Save your work and preview the form in your browser
6	Close all open windows.



## Working with Actions

- **Topic A:** Create Actions
- **Topic B:** Create Shared Actions
- **Topic C:** Control the Display of Actions
- **Topic D:** Create Action Hotspots



## Topic A: Create Actions

After completing this topic, you should be able to:

- ✓ Identify characteristics of and uses for actions.
- ✓ List the tasks associated with designing actions.
- ✓ List programming options for creating actions.
- ✓ Describe where and when actions are available to users.
- ✓ Identify where actions are displayed in the IBM® Lotus® Domino Designer® interface.
- ✓ Describe the properties you can set for actions.
- ✓ Insert system actions.
- ✓ Identify the characteristics of and uses for @commands.
- ✓ Describe the order of evaluation for @commands.
- ✓ Create actions.

### Actions

---

An **action** is a program associated with either a view or form and used to automate tasks for users. Actions are added to applications in order to:

- Automate repetitive tasks.
- Update information.
- Simplify the usage of the database.
- Perform complex calculations.
- Check for errors.

You can use an action to:

- Re-create IBM® Lotus Notes® menu commands as buttons. For example:
  - Create a document
  - Edit a document
  - Save a document
- Mail the current document.
- Calculate complex formulas that do not need to be saved with individual documents.

## Checklist: Designing Actions

Before creating an action, you should consider the following checklist.

	Task	Procedure
<input type="checkbox"/>	1	Define the purpose of the action.
<input type="checkbox"/>	2	Define the most appropriate programming language for the action.
<input type="checkbox"/>	3	Define where the action appears.
<input type="checkbox"/>	4	Define when the action appears.
<input type="checkbox"/>	5	Define the action's formatting.

## Programming Options Available In Actions

Like other design elements, you can program actions using one of the following options:

- Simple actions
- Formulas
- LotusScript
- JavaScript



**Note:** Simple actions cannot be customized and are not supported in Web applications.

## Action Display

Actions in a view are available to users when the view is opened. Actions in a form are available when a document created with the form is displayed, or a new document is created with the form.

Users access actions either from:

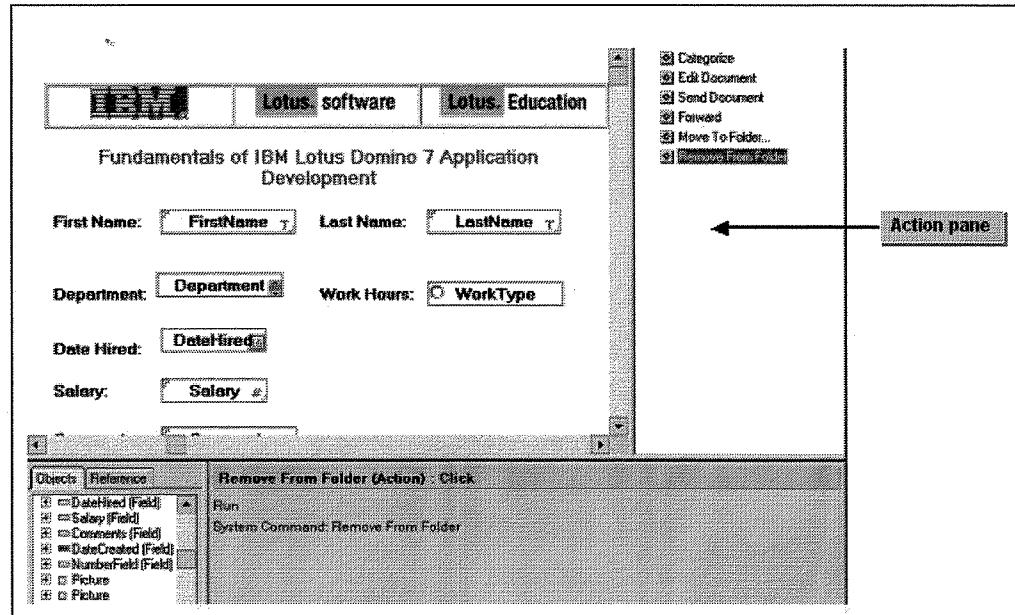
- The Lotus Notes Actions menu command
- An Action Bar button

The Action Bar is a row of buttons across the top of a form or view. To execute an action from the Action Bar, users simply click the button.

## The Action Pane

The Action pane displays the actions available to a particular form or view. You can select an action from the Action pane in order to open, edit, or delete the action.

The following figure shows the Action pane for a form.

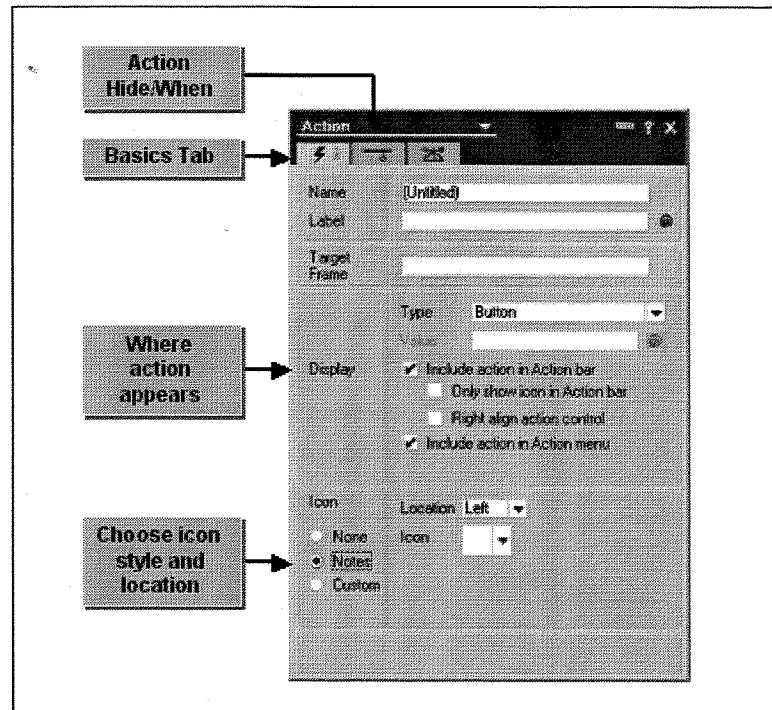


**Figure 12-1:** A form's Action pane

## Action Properties

Like other Lotus Domino Designer features, actions are modified using their properties box.

The following figure shows the Action Properties box.



**Figure 12-2: The Action Properties box**



## Activity 12-1: Create an edit action

To create a button on the Employee Information form that will allow users to edit documents, complete the following steps.

Step	Action
1	Open the <b>Employee Information</b> form in your copy of the Practice database in Lotus Domino Designer.
2	Choose <b>Create</b> → <b>Action</b> → <b>Action</b> from the menu.
3	In the <b>Action</b> Properties box, name the action <b>Edit Employee Info</b> .
4	If necessary, select the following display options: <ul style="list-style-type: none"> <li>● Type: <b>Button</b></li> <li>● <b>Include action in Action Bar</b></li> <li>● <b>Include action in Action menu</b></li> </ul>
5	Select the following hide/when options. Hide action when document is: <ul style="list-style-type: none"> <li>● <b>Previewed for editing</b></li> <li>● <b>Opened for editing</b></li> </ul>
6	Ensure that <b>Client</b> and <b>Formula</b> are selected in the Programmer's pane.
7	In the Script area of the Programmer's pane, type the formula: <code>@Command([EditDocument])</code>
8	Save the form and test the action on an existing document.

## System Actions

By default, new forms and views do not contain actions. Lotus Domino Designer provides a set of pre-defined system actions for common tasks. System actions can be added to a form or view.

The following is a list of IBM® Lotus® Domino® system actions:

- Categorize
- Edit Document
- Send Document
- Forward
- Move to Folder
- Remove from Folder

System actions cannot be modified and do not translate to Web applications. You can create your own actions to perform the same functions using the Formula language instead.

## Inserting system actions

Follow these steps to insert system actions into a form or view.

Task	Procedure
1	Open the form or view in Lotus Domino Designer.
2	Choose <b>Create</b> → <b>Action</b> → <b>Insert System Actions</b> . <b>Result:</b> The system actions are added to the Action pane.
3	Save the form or view.

## @Commands

**@Commands** are special functions that simulate the functionality of Lotus Notes menu commands. For example, use `@command ([Compose])` to create a button that creates a new document based on a specific form.

The syntax for using @commands is:

`@Command ( [command] ;parameters ) ;`

### Commonly used @commands

The following table contains examples of commonly used @commands.

@Com-mand	Function	Example
Compose	Creates a new document with the specified form.	<code>@Command ( [Compose] ; "Response" )</code>
FileSave	Saves the current document.	<code>@Command ( [FileSave] )</code>
FileCloseWindow	Closes the current window.	<code>@Command ( [FileCloseWindow] )</code>

## Topic A: Create Actions

### Lesson 12 ■ Working with Actions

@Command	Function	Example
EditDocument	Opens the currently selected document for editing.	@Command ([EditDocument])



**Caution:** To Web-enable all action buttons in a database as well as certain @commands, select the database property **Web access: Use JavaScript when generating pages**. Without this property set, Lotus Domino recognizes only the first button in a document and treats it by default as a Submit button that closes and saves the document.

### Order of Evaluation for @Commands

Lotus Domino evaluates formulas from top to bottom and left to right, completing each statement before proceeding to the next. @Commands are evaluated in one of two ways: as encountered or after all @functions. In most cases, the @command runs in the order specified in the formula.

For example, the following is an abbreviated list of @command functions that are evaluated as encountered or immediately.

Evaluated After All @Functions	Evaluated Immediately
EditClear	Clear
EditProfile	EditProfileDocument
FileCloseWindow	CloseWindow
FileExit	ExitNotes
ViewChange	SwitchView

### Creating Instant Messaging Chat Action Buttons

Lotus Domino Designer includes @functions and @commands that developers can use to integrate instant messaging into their Lotus Domino applications. The following table describes these instant messaging-related @functions and @commands.

@Function or @Command	Description
@GetIMContactListGroupNames	Returns the group names in the instant messaging contact list.
@Command([AddToIMContactList]) ...	Adds a name or names to a personal group in an instant messaging contact list.
@Command([SendInstantMessage])	Initiates a chat with one or more users.
@Command([ShowHideIMContactList])	Toggles the on/off display of the instant messaging contact list.

The following figure shows instant messaging action buttons displayed in a view.

Question	Who's Online?
▼ Holidays	Student ND01
▼ Holiday Calendar - 12/28/2005	Student ND01
One to two years ahead of time - 12/28/2005	
▼ Alternative Holidays - 12/28/2005	Student ND01
Selection of Alternative Holidays - 12/28/2005	
▼ Office Guidelines	
▼ Cubicle Decorations - 12/27/2005	Student ND01
Use map pins. - 12/28/2005	

**Figure 12-3:** Instant messaging action buttons in a view



## Optional Practice Activity 12-2: Create Instant Messaging Action Buttons in a View

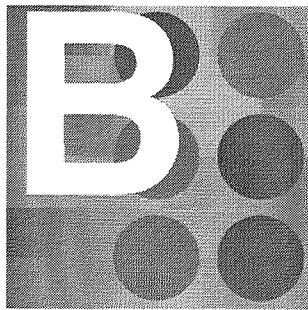


**Scenario:** The Worldwide Corporation would like to add buttons to the All Documents view in the Policies and Procedures database to provide users with the ability to contact document authors and to show/hide online status icons. Follow the steps below to modify the database.



**Note:** The guided solution to this activity can be found in Appendix A, Solutions to Practice Activities.

1. In the Q&A view of the Policies and Procedures database, create an instant messaging chat action button called Chat.  
Use the @command `SendInstantMessage`.
2. Create a second instant messaging chat action button called Show/Hide Contacts, using the @command `ShowHideIMContactList`.
3. In the Lotus Notes client, open the Q&A view, and confirm that the buttons are present and functional.



## Topic B: Create Shared Actions

After completing this topic, you should be able to:

- ✓ Identify characteristics of shared actions.
- ✓ Create shared actions.
- ✓ Determine when to use a shared action.

### Shared Actions

Shared actions are actions that you create as resources for the entire database. Once you create a shared action, you can insert it in multiple views and forms. If you modify the action, the modification applies to every form or view that refers to it.

### Creating a shared action

Follow these steps to create a shared action.

Task	Procedure
1	Open the database in Lotus Domino Designer.
2	In the Design pane, click <b>Shared Code</b> . Then, click <b>Actions</b> .
3	Click <b>New Shared Action</b> . The <b>Shared Actions Properties</b> box opens.
4	Create the shared action as you would any action.
5	Save the shared action.

## **Shared Action Properties**

---

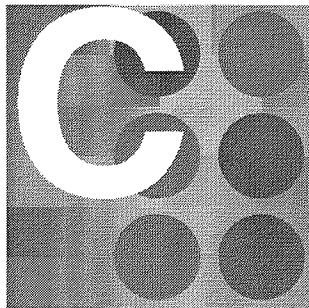
A shared action has the same properties as a standard action. You set the properties when you create the shared action. These properties determine how the action will behave in every form or view that uses it. The only property you can change on each form or view is the action's position on the Action Bar.

### **Inserting a shared action on a form or view**

---

Follow these steps to insert a shared action on a form or view.

<b>Task</b>	<b>Procedure</b>
1	In Lotus Domino Designer, open the form or view where you want to use the shared action.
2	If you are designing a view, choose <b>Create</b> → <b>Insert Shared Action</b> from the menu. If you are designing a form, choose <b>Create</b> → <b>Action</b> → <b>Insert Shared Action</b> .
3	Select the shared action you want to insert. Click <b>Insert</b> .
4	<b>Click Done.</b> <b>Result:</b> The shared action is inserted into the <b>Action</b> pane.
5	Preview the form or view, and test the action.



## Topic C: Control the Display of Actions

After completing this topic, you should be able to:

- Identify the characteristics of subactions.
- Describe the order in which actions are displayed.
- Describe how to hide action buttons.
- Identify characteristics of the Action bar.
- Describe how Web clients display actions.

### Sub Actions

There are a couple of ways to enhance the display of actions. This section examines how to:

- Create actions with sub actions.
- Customize the Action bar.

Sub actions group related actions under a single main menu choice or button.

Use sub actions if you have:

- A long list of actions and do not want to overwhelm users
- Related actions that you want to group together
- Very little space left in the action bar and want to avoid requiring the user to scroll horizontally.

### Adding sub actions to a form or view

Follow these steps to add sub actions to a form or view.

Task	Procedure
1	Open the form or view you want to modify.
2	Choose <b>Create→Action→Action with Sub Action</b> . <b>Result:</b> An untitled action is created as a placeholder for the new sub action. To name this action placeholder, open its Properties box.
3	Enter a name for the sub action.
4	Program the sub action.
5	Save and preview the form or view.
6	To add more sub actions, select the placeholder, then choose <b>Create→Action</b> .

## **Action Order**

---

Actions and sub actions appear in the order in which they were created. You can rearrange this order in a more logical sequence.

### **Rearranging action order**

---

Follow these steps to rearrange the order of actions and sub actions.

<b>Task</b>	<b>Procedure</b>
1	Open the form or view with actions in Lotus Domino Designer.
2	Open the Action pane.
3	Click the action or action header.
4	Click and drag the action to a different position.
5	Save and preview the form or view.

## **Hiding Actions and Action Buttons**

---

As with other form or view elements, you can conditionally restrict the availability of an action using the action's Hide/When properties. These are the same Hide/When properties available to other design elements. Use these properties to make an action visible when it is relevant.

For example:

- Hide an action that edits a document when the document is in **Edit** mode.
- Hide an action that saves a document when the document is in **Read** mode.



**Note:** Hide/When formulas execute when the document or view with which the action is associated is refreshed.

## Hiding action buttons

Follow these steps to hide an action or action button.

Task	Procedure
1	Open the form or view.
2	Select the action.
3	Choose <b>Design</b> → <b>Action Properties</b> .
4	Click the <b>Action Hide When</b> tab.
5	Select the appropriate options.
6	Save and preview the form or view.

## Action Bar

The Action Bar displays action buttons at the top of a form or view. Lotus Domino gives you the option of changing some of the Action Bar's properties.

To alter the Action Bar, you can:

- Change the Action Bar's appearance by specifying:
  - Alignment options for action buttons
  - Background image or color for buttons on the bar
  - The style and color of the bottom border
- Enable the Lotus Domino Action Bar applet for Web users.

Actions that appear on the button bar use the **Action Bar** Properties box to set additional characteristics, such as font, color, and button size.

## Changing Action Bar Properties

Follow these steps to enhance the look of an Action Bar.

Task	Procedure
1	Open the form or view in Designer.
2	Select any action in the Action pane.
3	Choose <b>Design→Action Bar Properties</b> from the menu. <b>Result:</b> The <b>Action Bar Properties</b> box opens.
4	On the <b>Action Bar Info</b> tab: <ul style="list-style-type: none"><li>Select the button bar alignment.</li><li>Select how to display the button bar for Web users.</li></ul>
5	On the <b>Action Bar Size</b> tab: <ul style="list-style-type: none"><li>Select the button bar size.</li></ul>
6	On the <b>Action Bar Background</b> tab: <ul style="list-style-type: none"><li>Select the button bar background color.</li><li>Select the button bar background image and how it is displayed.</li></ul>
7	On the <b>Action Bar Border</b> tab: <ul style="list-style-type: none"><li>Select the button bar border style and effect.</li></ul>
8	On the <b>Button Properties</b> tab: <ul style="list-style-type: none"><li>Select button size.</li><li>Select button display options.</li><li>Select button background color or image.</li></ul> The changes affect all buttons on the Action Bar.
9	On the <b>Button Font</b> tab: <ul style="list-style-type: none"><li>Select the font properties for all buttons on the <b>Action Bar</b>.</li></ul>
10	Close the <b>Action Bar Properties</b> box.

## **Web Access**

---

In the Action Bar Properties box, you can use a Java applet for Web access. The Java applet:

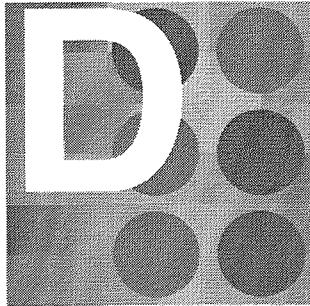
- Provides a scroll bar, when needed.
- Looks like the Action Bar in the Lotus Notes client.
- Supports cascading actions. When the user clicks the main action, the action selections appear as a second row of slightly smaller buttons.

## **Enabling the Action Bar applet**

---

Follow these steps to enable the Action Bar as an applet.

<b>Task</b>	<b>Procedure</b>
1	Open the <b>Action Bar Properties</b> box.
2	In the <b>Basics</b> tab, select <b>Using Java Applet</b> under <b>Web Access Display</b> .
3	Close the <b>Action Bar Properties</b> box.



## **Topic D: Create Action Hotspots**

After completing this topic, you should be able to:

- ✓ Identify the characteristics of hotspots.
- ✓ Create action hotspots.

### **Hotspots**

Action hotspots are programmable areas on a form, subform, or page that can complete an automated task.

Hotspots can be associated with:

- Text
- Graphics

### **Creating an action hotspot**

Follow these steps to create an action hotspot.

<b>Task</b>	<b>Procedure</b>
1	Open a form, subform, or page.
2	Click and drag to select the text or image to be associated with a hotspot.
3	Choose <b>Create</b> → <b>Hotspot</b> → <b>Action Hotspot</b> . <b>Result:</b> An Action Hotspot Properties box appears.
4	In the <b>Run</b> area of the Programmer's pane, select either <b>Web</b> or <b>Client</b> from the drop-down list.
5	Select a programming language from the code drop-down list.
6	Enter the code in the Programmer's pane.
7	Save and preview the form, subform, or page.



## Practice Activity 12-3: Add Automation to the Policies and Procedures Application



**Scenario:** Worldwide Corporation would like to automate creating and editing documents in the Policies and Procedures application. Users should be able to:

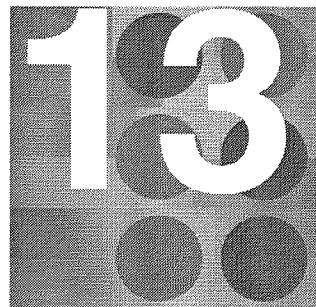
- Create a policy or a question from the Policies view.
- Edit a policy when the document is open.
- Create a question or an answer from the Q&A view.
- Edit a question or an answer either when a document is open or selected in the Q&A view.
- Save any of the documents by clicking a button instead of using menu commands.

Perform the following tasks to complete this activity.



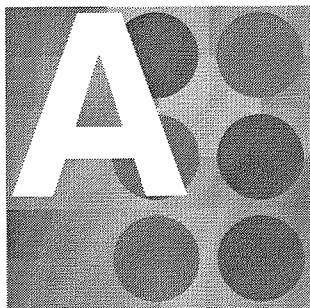
**Note:** A guided solution to this activity can be found in Appendix A, Solutions to Practice Activities.

1. Create a categorized Policies view, showing Category Policy Number, title, and Effective Date.
2. Make a list of actions you need to create and where they are to be used.
3. Determine which of them could be shared.
4. Create the actions and add them to the appropriate views and forms.
5. Hide the actions at the appropriate times. For instance, if a document is in Edit mode, there is no reason for the Edit action to be available.
6. Close all open windows.



## Working with Agents

- **Topic A:** Investigate Agents
- **Topic B:** Create Agents



## Topic A: Investigate Agents

After completing this topic, you should be able to:

- ✓ Identify the characteristics of and uses for agents.
- ✓ Describe the properties associated with agents.
- ✓ List design considerations for agents.
- ✓ List programming options available for creating agents.
- ✓ Identify the characteristics of and uses for the SELECT statement.
- ✓ Define the FIELD keyword and describe its use.
- ✓ Describe design options for activating agents.
- ✓ Describe agent events and their triggers.
- ✓ List options for scheduling events.
- ✓ Identify the relationship between agents and the documents they can process.

### Agents

---

**Agents** are programs within an IBM® Lotus® Domino® application that perform a series of automated tasks. The benefits of using agents include:

- Agents can run manually by users or automatically in the background as scheduled agents.
- Agents are not associated with a specific design element.
- Agents can be run on a specific server, on several servers, on workstations, or on the Web.
- Agents can call other agents.
- Agents can consist of simple actions, formulas, LotusScript, or Java programs.
- Agents can be distributed easily because they can be replicated.
- Agents can be shared or private.

You can use agents to set up user-activated or background tasks that run on a schedule. You can use agents to run automated tasks for views, documents, fields, and databases.

## Actions versus agents

Agents are different from actions in that:

- Form and view actions work only when the form or view is open.
- Agents are not restricted to a specific form or view. Use agents to set up user-activated tasks or background tasks in any part of a Lotus Domino application.

## Examples of agents

Use agents to:

- Update a group of documents.
- Archive documents.
- Send messages and documents.
- Run other agents.

## Types of agents

There are two types of agents:

- **Shared** agents are agents created by one user and can be run by other users or scheduled to run on the server.
- **Private** agents are agents that users create for themselves. Users cannot run another user's private agent. Private agents are stored on the machine where the agent is created.

Once an agent is created, its type can be changed from shared to private, and vice versa, at a later date. This could be used to test a new agent on a "live" application before it is distributed generally.

## Parts of an agent

The following table describes the parts of an agent.

Part	Function
Trigger	Specifies when the agent program should run.
Target	Selects documents the agent program acts upon.
Program	Specifies the code to execute.

## The Agent Work pane

The following figure shows the Agent Work pane and the Agent Properties box.

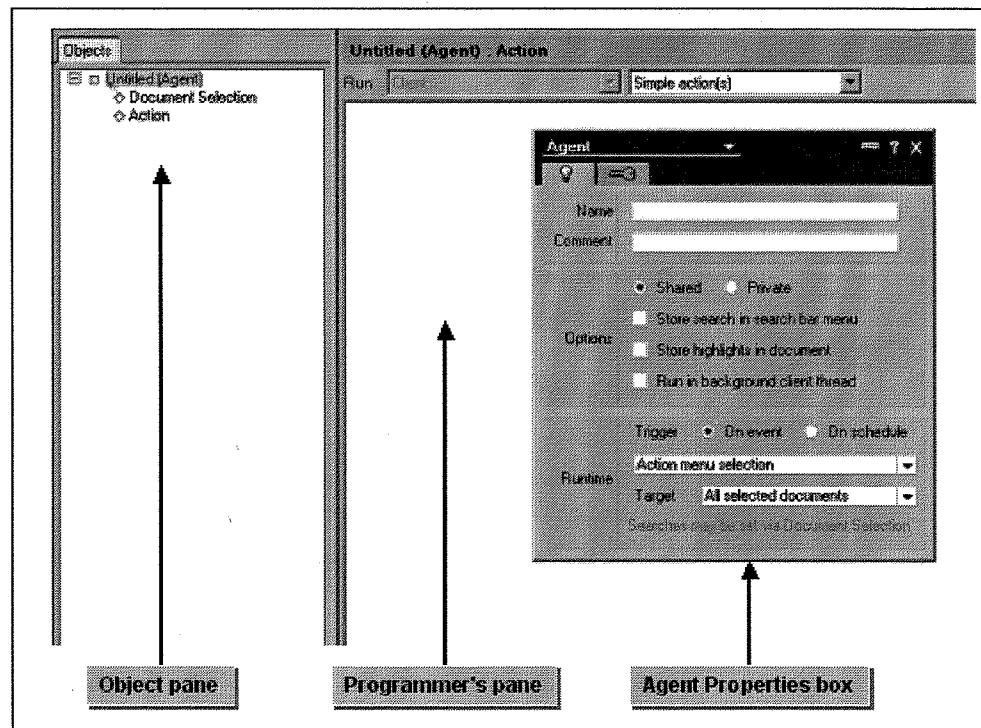


Figure 13-1: The Agent Work pane and the Agent Properties box

## Agent Properties

Use options in the Agent Properties box to define an agent to run:

- Manually or automatically.
- Against all documents or a subset of documents in a database.

## The Agent Properties box

The following figure identifies parts of the Agent Properties box.

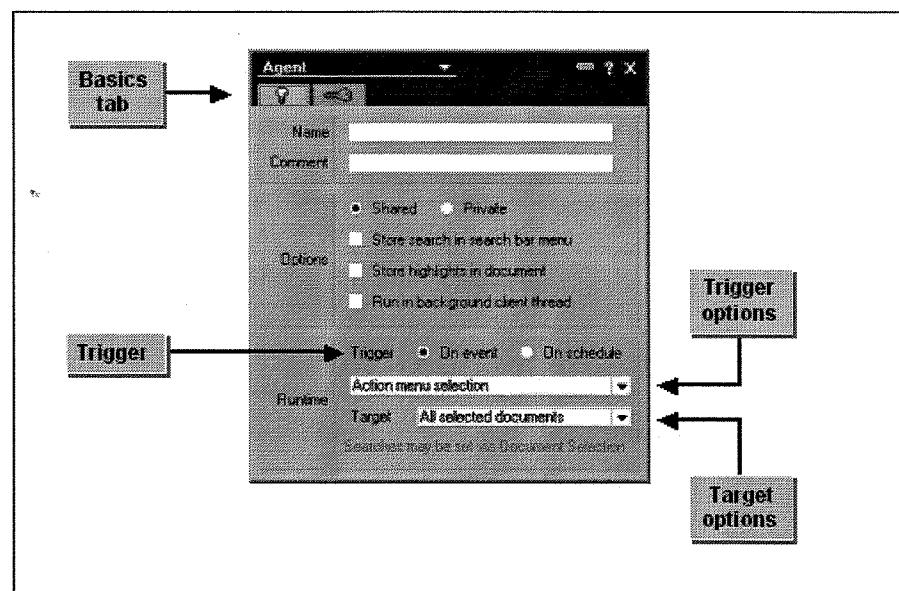


Figure 13-2: The Agent Properties box

## The Agent Basics Tab

In addition to naming an agent and determining its trigger and target, the Basics tab of the Agent Properties box has a few more options.

The following table lists these options and their functions.

Option	Function
Store search in search bar menu	Displays the search query in the search bar.
Store highlights in document	Search matches are highlighted in the searched documents.
Run in background client thread	<p>Runs an agent in the background without having to wait until it completes.</p> <p>Note that you should check this option only if the agent (or the top-level agent that calls this agent) is triggered by "On event - Action menu selection."</p> <p>To get the full benefit of this feature, the database should reside on a server and the agent must be invoked via the client.</p>

## **Designing an agent**

---

Before creating an agent, you should consider the following checklist.

<b>Task</b>	<b>Procedure</b>
1	Decide what the agent is to accomplish.
2	Decide when it should run.
3	Define what documents it should run against.
4	Decide where the agent is going to run.

## **Programming Options**

---

Agents can run any of the following types of programs:

- Simple functions
- Formula
- LotusScript
- Java

You can program an agent to do almost anything in an application.

### **Common formula agent tasks**

Two common tasks in formula agents are to:

- Refine which documents the agent processes.
- Update fields in documents.

## **The SELECT Statement**

---

Formula-based agents run iteratively on the target documents. The number of documents that an agent processes directly affects the time it takes to execute the agent's formula. To limit the target documents, use the SELECT statement.

The syntax for using SELECT is as follows:

```
SELECT logicalexpression
```

The following example selects only documents created using the Employee Information form:

```
SELECT Form="Employee Information"
```

If you do not include a SELECT statement in the formula, IBM® Lotus® Domino Designer® appends the following formula to your code:

```
SELECT @All
```

Generally, a SELECT statement must be the first statement in a formula. SELECT @All is the only one that can be placed at the end of a formula.

## **Field Updates**

---

Agents can add new fields to existing documents or modify existing fields. To do this, use the keyword FIELD in a formula. FIELD is a reserved word used to add a value to a new or existing field.

The syntax for using FIELD is as follows:

```
FIELDfieldname := value
```

The following example changes the LastName field value to Sunshine:

```
FIELD LastName := "Sunshine"
```



**Note:** If the LastName field does not exist in the document, it is created and given the value Sunshine. The field does not become visible unless you add it to the form design, but you can access it by naming it in formulas.

## Agent Activation

Once you have decided what the agent will do, you need to determine how you want it activated.

The following table describes design options for when the agent runs.

Agents Can Run...	Example
Manually	By the user via the <b>Actions</b> menu in the Lotus Notes client
When triggered by an event	If documents have been created or modified
According to a schedule	Every day at 1:00 AM

## Agent Events

The following table describes the events that can trigger an agent and examples of when to use them.

Event	When to Use It
Before new mail arrives	Before mail is deposited in the mail database; for example, to delete unwanted mail.
After new mail arrives	Once new mail is delivered to the client; for example, to forward mail while a user is on vacation.
After documents are created or modified	To update new or existing documents; for example, to change an area code for phone numbers.
When documents are pasted	When copied documents are pasted into the database and need to be modified as they are being pasted; for example, to add the current date to the document.
Action menu selection	The agent is manually activated by the user or is a Web Query Save or Web Query Open agent.
Agent list selection	The agent is run only by another agent or is still under development and you need to hide it from the users.

## Scheduled Events

The following table lists when agents can be scheduled to run.

Schedule	Use for...
More than once a day	Critical tasks that are run frequently. Any agent, regardless of how frequently it is run, can severely affect a server's performance.
Daily	Important tasks that are not as high priority as critical tasks; for example, sending a company-wide e-mail.
Weekly	Routine tasks; for example, calculating a department's sales performance.
Monthly	Low-priority tasks; for example, deleting older documents.
Never	Agents that you do not want to run in particular circumstances; for example, agents that run on the Web or agents that are disabled or called by other agents.



**Note:** In most circumstances, you should schedule the running of agents based on the business need for having the work accomplished in a proper time frame.

## Document Processing

Options for which documents the agent processes depend on when the agent runs.

The following table shows examples of which documents different agents can process.

An Agent that Runs...	Can Process...
Manually from the <b>Actions</b> menu	<ul style="list-style-type: none"> <li>● Documents that users select</li> <li>● All documents in a view</li> <li>● All documents in the database</li> </ul>
When triggered by an event	<ul style="list-style-type: none"> <li>● Only new and modified documents</li> </ul>

## Topic A: Investigate Agents

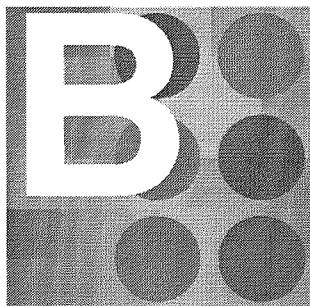
---

### Lesson 13 ■ Working with Agents

An Agent that Runs...	Can Process...
According to a schedule	<ul style="list-style-type: none"><li>● All documents in the database</li><li>● New documents and documents modified since the last run of the agent</li></ul>



**Note:** SELECT statements in Formula language, as well as other statements in LotusScript and Java, can modify all documents in the database.



## Topic B: Create Agents

After completing this topic, you should be able to:

- ✓ Describe agent names.
- ✓ Generate agents.

### Agent Names

Agent names within the same database need to be unique. However, if the same agent is used in several different databases, use the same agent name to help users recognize the agent's purpose.

Agent names:

- Are case-sensitive.
- May include letters, numbers, spaces, and punctuation.
- Should be self-descriptive.
- Should be short, if appearing as items on the Actions menu.

The names you give to manually run agents appear as options on the Actions menu. Use an alias (an internal name for an agent) when naming an agent. This allows you to change or translate the name that users see without disabling formulas that reference the original name.

### Creating an agent

Create the agent after you have defined what your agent should do as well as when and where it should run.

Follow these steps to create an agent.

Task	Procedure
1	Open the database in Lotus Domino Designer.
2	Select <b>Agents</b> under the <b>Shared Code</b> design element.
3	Click <b>New Agent</b> .
4	Name the agent and add a comment, if desired.
5	Select the agent option <b>Shared or Private</b> .
6	Select the agent trigger <b>On Event</b> or <b>On Schedule</b> .
7	Select the appropriate agent trigger option from the drop-down list.
8	Select the appropriate documents for the agent to target from the <b>Target</b> option drop-down list.

## Topic B: Create Agents

---

### Lesson 13 ■ Working with Agents

Task	Procedure
9	Close the <b>Agent Properties</b> box.
10	Select the appropriate programming type from the drop-down list.
11	Enter the program's code into the Programmer's pane.
12	Save and test the agent.



## Practice Activity 13-1: Update Existing Documents



**Scenario:** So far, you have either left the policy number blank on Policy documents, or made one up. In this practice activity, you will write an agent that computes a policy number for all existing policies. The policy number should be formatted as “PN999999” where the characters “PN” are the first two characters in the policy number, and “999999” is calculated based on the rightmost six characters of the document’s unique ID.



**Note:** The guided solution to this activity can be found in Appendix A, Solutions to Practice Activities.

Perform the following tasks to complete this practice activity:

1. Write a formula that computes the policy number and saves the new policy number in the PolicyNo field.

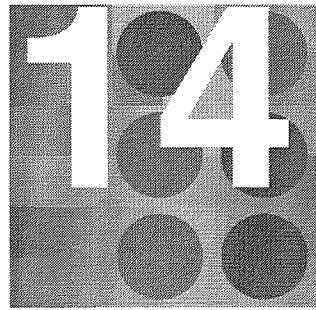


**Hint:** Look up the @DocumentUniqueID.

2. Make the agent run on all existing policy documents.
3. Run the agent and check the results.
4. **Challenge**

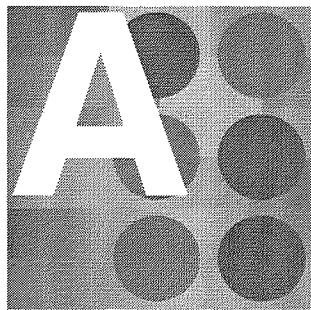
Modify the Policy form so that it computes a new policy number when a new Policy document is created.

5. Close all open windows.



## Securing Applications

- **Topic A:** Explore Database Access
- **Topic B:** Implement Roles
- **Topic C:** Use Readers and Authors Fields
- **Topic D:** Control Access for Anonymous Users



## **Topic A: Explore Database Access**

After completing this topic, you should be able to:

- ✓ Describe general database security measures.
- ✓ Identify the characteristics of access control lists.
- ✓ Identify access control levels.
- ✓ Describe the user identifiers in an access control list.
- ✓ Describe examples of defining access levels for certain situations.
- ✓ List the steps for displaying a database's access control list.
- ✓ Describe the available access options.
- ✓ Describe user types.

### **Database Security Overview**

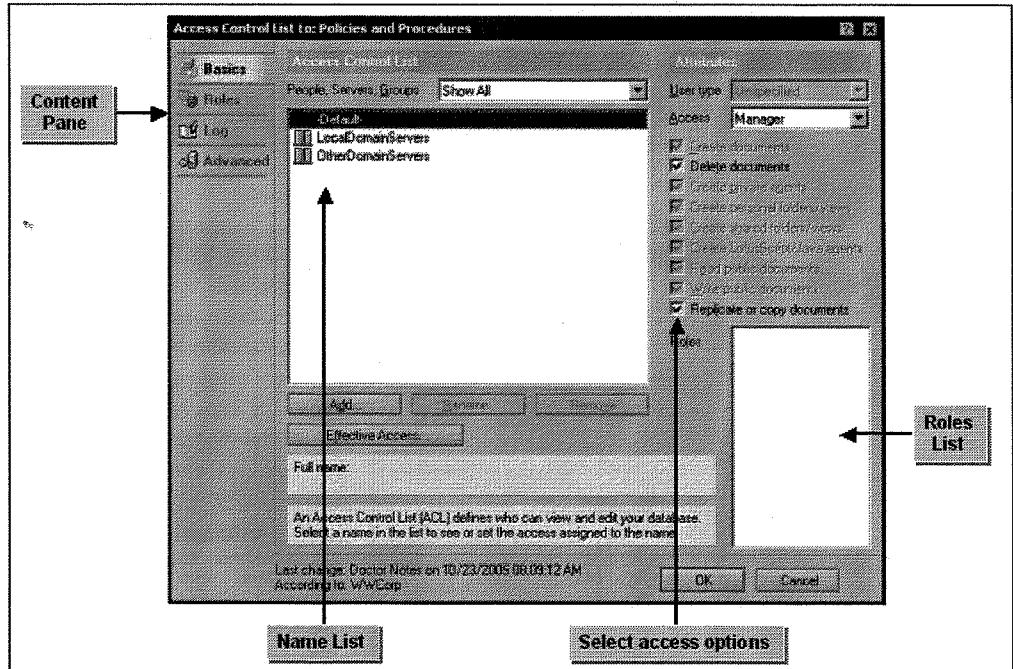
Until this point, the focus of your database design has been on building an application that meets the needs of the users. Now that the process is near completion, database security becomes more critical. This includes securing the database from unauthorized entry as well as restricting user access within the database. As you learned earlier in the course, the layered security model of IBM® Lotus® Domino® provides many ways to customize security to the database and its content.

Access restriction begins at the network level. Once a user has been authenticated at this level, the next restriction is at the server level. Typically, network administrators and system administrators configure and maintain these two levels. Once a network and server admit a user, that user encounters access restrictions or controls at the database level. Typically, Lotus Domino database managers configure and maintain this level using **Access Control Lists (ACLs)**.

### **Access Control Lists**

Every Lotus Domino database includes an Access Control List. Manager access is required to modify the ACL. Managers of a Lotus Domino database add users (individually or in groups) to the ACL and assign them a certain access level that entitles them to work within the application. When users attempt to open a database, Lotus Domino refers to the Access Control List to determine their privileges within the database.

The following figure shows the Access Control List dialog box.



**Figure 14-1: The Access Control List dialog box**

### The Access Control List dialog box

There are four panels in the Access Control List dialog box. The following table lists each panel and its function.

This Panel...	...Is Where the Domino Database Manager
Basics	Assigns users, groups, and servers their levels of access.
Roles	Creates user roles.
Log	Views a history of the last 20 changes made to the Access Control List. It also tracks who made the changes and what was done.
Advanced	Performs advanced security tasks.

## Access Control Levels

The following table lists each access control level and the privileges it gives to a user.

Access Control Level	Privileges
No Access	No access to the database at all.
Depositor	Depositors can create documents, but cannot read, edit, or delete documents, including those they create.
Reader	Readers can read documents, but cannot create, edit, or delete documents.
Author	Authors can create and read documents, but cannot edit documents unless specified in the document itself.
Editor	Editors can create, read, and edit all documents unless there are restrictions on specific documents.
Designer	Designers can perform the same functions as an Editor, except where restrictions exist on specific documents. Designers also can modify the database design, but cannot delete the database or change its ACL.
Manager	Managers can perform all functions, including creating and modifying the ACL and deleting the database.

## User Identifiers

The **Lotus Domino Directory** is like a phone book of all the users and servers in the Lotus Domino environment. System administrators set up user names and groups of users that developers can use when defining access to applications. The database manager adds individual user names, group names, and server names to the ACL of the database to control access to the application.

The Access Control List defines access rights for users in four ways. The following table describes the purpose of each option.

Option/ Description	Defines Access for...
Default	Any user who is not specifically listed in the ACL (individually or in a group).
Groups	A list of people who can all access the same functionality in the database. If a user is a member of two groups, and both groups appear in the ACL, the user receives the higher access level.
Individual names	A specific IBM® Lotus Notes® or Web user. This will override settings for any group to which the individual belongs.
Anonymous	Any unauthenticated Lotus Notes or Web user. If there is no Anonymous entry in the Access Control List, unauthenticated users will use the Default access level.

## **User Access Levels**

The following are some examples of setting user access levels.

In Order to..	Set...
Prevent unknown users from accessing the database	Default to <b>No Access</b> .
Grant anyone the ability to read documents	Anonymous to the <b>Reader</b> access.
Allow all users in the Students group to modify the application	Students to <b>Designer</b> access.
Preview your forms in a Web browser, as you are working on the database design	Anonymous to <b>Author</b> access.

## **Displaying a Database's Access Control List**

Any user with access to a database is able to see its ACL.

## **Displaying a database's ACL**

---

Follow these steps to display the ACL of a database

<b>Task</b>	<b>Procedure</b>
1	Select or open the database in either the Lotus Notes client or IBM® Lotus® Domino Designer®.
2	Choose <b>File</b> → <b>Database</b> → <b>Access Control</b> from the menu. <b>Result:</b> The <b>Access Control List</b> dialog box appears.

## **Access Options**

---

Each access level has up to eight options that Lotus Domino database managers can turn on or off, further refining users' privileges. For example, you can control whether users with Editor access can delete documents.

The following table describes the default and optional access privileges and which levels they are associated with.

Access Level	Automatic Privileges	Optional Privileges
No Access	None	Read public documents Write public documents
Depositor	Create documents	Read public documents Write public documents
Reader	Read public documents	Create private agents Create personal folders/views Create LotusScript/Java agents Write public documents
Author	Read public documents	Create documents Delete documents Create private agents Create personal folders/views Create LotusScript/Java agents Write public documents
Editor	Create documents Read public documents Write public documents	Delete documents Create private agents Create personal folders/views Create shared folders/views Create LotusScript/Java agents
Designer	Create documents Create private agents Create personal folders/ views Create shared folders/views Read public documents Write public documents	Delete documents Create LotusScript/Java agents
Manager	Create documents Create private agents Create personal folders/ views Create shared folders/views Create LotusScript/Java agents Read public documents Write public documents	Delete documents

Enable the read and write public documents options to give users with No Access or Depositor access limited access to specific forms, views, and documents.

## User Types

User types specify whether a name in the ACL is a person, server, or group. They can protect your database from the following scenario.

- Chris Jones opens the ACL to the Policies and Procedures database and notes that Terry Smith has Editor access.
- Chris Jones gets access to the server, creates a group called Terry Smith, and adds himself to the group. Chris Jones now has access to the Policies and Procedures database as part of the group Terry Smith.

One way to prevent this is to designate Terry Smith as an individual with the Person user type.

The following table lists each user type, to whom it can be assigned, and what it does.

User Type	Assigned to...	Description
Person	An individual user, including individuals on a server workstation	Enforces access as a single user.
Server	A single server, including a server console and server workstation	Prevents someone from accessing the database from a Lotus Domino workstation using the server ID.
Server Group	A group of servers	Identifies a group of servers that will host replicas of the database.
Person Group	A group of individual users	Grants Person access to each user in a group without listing each user in the ACL.
Mixed Group	A group of servers and individual users	Grants the same access to a group of users and servers.
Unspecified	Any user, group, or server	Has Lotus Domino look up the user type in the Address Book.

The database manager applies user types to meet the access restrictions.

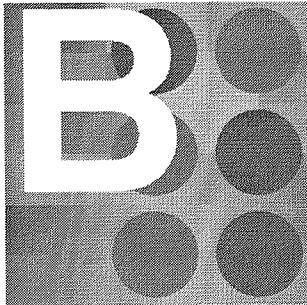


## Practice Activity 14-1: Set Access Restrictions to the Database



**Scenario:** Worldwide Corporation's Policies and Procedures application is almost ready to be deployed. However, security levels have not been established for all the different types of users. In this group activity, you will work with the rest of the class to determine the access control levels required for the appropriate users.

1. Working with the instructor and the rest of the class, plan the ACL.  
**For the Default user type, the \_\_\_\_\_ access control level should be used.**
2. **For the Anonymous user type, the \_\_\_\_\_ access control level should be used.**
3. **For the Employees group, the \_\_\_\_\_ access control level should be used.**
4. **For the Policy Makers group, the \_\_\_\_\_ access control level should be used.**
5. **For the Department Head, the \_\_\_\_\_ access control level should be used.**
6. **For the Managers group, the \_\_\_\_\_ access control level should be used.**
7. **For the Designer, the \_\_\_\_\_ access control level should be used.**
8. Observe as your instructor modifies the ACL for the Policies and Procedures database, and saves the file as **PoliciesACL.nsf**.
9. Make a new copy of the Policies and Procedures database in the Fundamentals folder on the classroom server (Hub/WWCorp). Append your initials to the end of the database name and the file name. (Example: If your initials are ABC, name the database **Policies and Procedures ABC**, and the file **Policies ABC.nsf**. You will use this copy later in this lesson.)
10. Set your access control level to Manager.



## **Topic B: Implement Roles**

After you complete this topic, you should be able to:

- ✓ Describe user roles.
- ✓ Describe the procedures for defining a role in the access control list.
- ✓ Describe how roles can be used in formulas to control access.
- ✓ List @functions that are commonly used to work with roles.
- ✓ Use a formula to set the Hide/When property for a field.

### **Roles**

Roles provide a way to define a group within the database. Users and groups are assigned roles to refine access to particular views, forms, documents, or sections. Instead of assigning access to specific users and groups, assign access to the role. Roles are functional after being:

- Defined in the ACL
- Used in a design element

### **Advantages of using roles**

Some of the advantages of using roles are that they:

- Provide a flexible method of restricting access to a specific set of users.
- Can be used in formulas.
- Provide a method to define a set of users within an application.
- Do not need to be maintained or added as a group in the organization's Lotus Domino Directory.
- Make it unnecessary to modify the application when users leave or new users join.
- Can be assigned to multiple groups and users in the ACL, which can simplify application development and maintenance.

### **Defining Roles**

Defining a role in the ACL is a two-step process:

1. Create the role in the ACL.
2. Assign roles to users or groups.

## Creating a role in the Access Control List

Follow these steps to create a role in the ACL.

Task	Procedure
1	Choose <b>File</b> → <b>Database</b> → <b>Access Control</b> . <b>Result:</b> The Access Control List box opens.
2	Click <b>Roles</b> in the Contents pane.
3	Click <b>Add</b> . <b>Result:</b> The <b>Add Roles</b> dialog box appears
4	Enter a name for the role. The name may be up to 15 characters long.
5	Click <b>OK</b> . <b>Result:</b> The Role name appears in brackets in the <b>Roles</b> list.

## Assigning roles to users

Once a role is created, assign it to the appropriate users or groups listed in the ACL. Follow these steps to assign roles to users or groups.

Task	Procedure
1	Open the database in either the Lotus Notes client or Lotus Domino Designer.
2	Choose <b>File</b> → <b>Database</b> → <b>Access Control</b> .
3	Highlight any entry in the <b>Name</b> list.
4	Select a Role from the <b>Roles</b> dialog box. <b>Result:</b> A check mark appears beside the Role selected.
5	Click <b>OK</b> . Once roles have been assigned to users, you can use them to determine what users can do or see in the database.

## Access Control

Incorporate roles into formulas to control access to parts of the application and to define responsibilities within the application. By using the name of the role in a formula, rather than actual user names, you simplify the construction of the formula and make the application easier to maintain.

You can apply roles to further refine control within the application. For example, with roles, you can control who can:

- Access specific views and folders
- Create documents
- Read the documents
- Edit the documents
- Access portions of a document

This section shows how to write a formula to determine who can see a specific design element based on a role. This involves capturing the current user's role and comparing it in a Hide When formula.

## @Functions and Roles

Use the following functions to capture the current user's role(s).

Function	Returns
@UserRoles	A text list of the roles for the current user
@UserNamesList	A text list including the current user name, any groups for the current user, and any roles for current user

## Comparing the User's Role

Once a user's role is determined, it needs to be compared against a list. The formula's course of action is based upon whether or not the role matches an entry on the list.

Use the @IsMember function to determine if a user has a specific role. This function determines if a text string matches an item in a text list. The syntax for using @IsMember is:

```
@IsMember(textValue; textListValue);
```

Use the @IsNotMember function to determine if a text string does not match an item in a list. The syntax for using @IsNotMember is:

```
@IsNotMember(textValue; textListValue);
```

The following table provides some examples of using @IsMember and @IsNotMember.

Example	Returns
<code>@IsMember("[Student]"; @UserRoles)</code>	True, if the current user is assigned the Student role
<code>!@IsMember("[Student]"; @UserNamesList)</code>	True, if the current user is not assigned the Student role
<code>@IsNotMember("[Student]"; @UserRoles)</code>	True, if the current user is not assigned the Student role

The following information is presented to support the previous table:

- All roles assigned through the database ACL begin and end with square brackets, as shown in the examples above.
- There is a system-assigned role, \$\$Web Client, that automatically appears whenever the user is connected to the database through a Web browser. It does not contain the square brackets. It can be used to affect screen presentation and logic in applications used with both the Lotus Notes client and Web browsers.
- Roles do not function on a local replica of a database unless the advanced ACL check box Enforce a consistent Access Control List across all replicas is selected.

## Using a formula to set the Hide/When property

---

Follow these steps to hide a field with a formula.

Task	Procedure
1	Open the design element's <b>Properties</b> box.
2	Click the <b>Paragraph Hide/When</b> tab.
3	Select <b>Hide paragraph if formula is true</b> .
4	Enter a formula that evaluates to True (1) or False (0).



**Caution:** Using Hide When to hide the contents of a field does not provide true security. Users can still see the contents of the field by using the **Document Properties** box. Use encryption to secure the data in a field for Lotus Notes clients.



## Practice Activity 14-2: Create Roles



**Scenario:** In this activity, you will create roles for the Policies and Procedures application and assign these roles to different users. For example, there are policy makers who should be able to view all documents and assign questions to the appropriate experts.



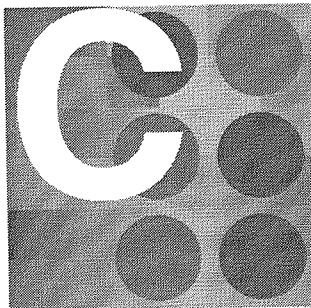
**Note:** The guided solution to this activity can be found in Appendix A, Solutions to Practice Activities.

Perform the following tasks to complete this activity.

1. Determine appropriate roles for the application. You will do this in conjunction with the instructor and the rest of the class.

Role	Description

2. Create the roles in your copy of the Policies and Procedures ABC database.



## **Topic C: Use Readers and Authors Fields**

After completing this topic, you should be able to:

- ✓ Describe document access control methods.
- ✓ Identify the characteristics of Readers fields.
- ✓ Create a Readers field.
- ✓ Identify the characteristics of Authors fields.
- ✓ Create an Authors field.

### **Document Access**

Individual documents sometimes contain sensitive information. Lotus Domino's layered security model provides mechanisms to restrict access to a document.

The following table lists what methods control who can create, read, and edit documents.

<b>Access</b>	<b>Controlled by</b>
Read access to documents	<ul style="list-style-type: none"><li>● Read access lists</li><li>● Readers fields</li></ul>
Edit access to documents	<ul style="list-style-type: none"><li>● Authors fields</li><li>● Controlled Access sections</li></ul>
Create access	<ul style="list-style-type: none"><li>● Form property: Show in Create menu</li><li>● Create access lists</li><li>● Action buttons</li></ul>

This section focuses on using Readers and Authors fields to control document access.

## Readers Fields

A **Readers field** restricts readership of a document to those users and servers that are listed in the field. It is a special type of multiple-value text field designed to interact with the Access Control List (ACL).

By specifically listing users, groups, and roles in a Readers field, you can control who can read individual documents. This can be very useful in a scenario requiring an approval process. With a Readers field, you can restrict access to any documents that are not approved; if the documents are approved, the restrictions are unnecessary.

### How Readers fields work with the ACL

Entries in a Readers field cannot give a user more access than what is specified in the ACL; they can only further restrict access. Users with No Access cannot gain access to a database even if their names are listed in a Readers field in a document in that database.

For example, users with Editor access or above in the ACL can be restricted from reading documents that have a Readers field if their names are not included in that field.

The names in Readers fields must be the full Notes hierarchical names (example: Doctor Notes/WWCorp) to prevent the possibility of two people with the same name (in different locations or containers) from being able to read the documents in question.



**Caution:** This type of access control always works when the database is on a server. However, it only works on a local database when the advanced ACL check box Enforce a consistent Access Control List across all replicas is selected.

### Viewing documents with Readers fields

When Lotus Domino builds a view, it indexes all documents that meet the selection criteria, regardless of whether those documents contain a Readers field. When that view is displayed:

- Documents that do not contain a Readers field are visible to all users.
- Documents with a Readers field are only visible to users listed in the Readers field. Other users do not see those documents.
- In a categorized view, categories that exist only in invisible documents will not appear if the row check box **Don't show empty categories** is selected. If the check box is not selected, the category will be displayed, but the documents associated with it will not appear.

Thus, you may not need to alter the view to hide certain documents from general viewing.

## **Specifying Readers**

---

Entries in a Readers field must compute to a text value. Designers and managers specify readers in a Readers field by using one of the following methods:

- Write a formula to compute user names, roles, or groups. This is the most frequently used option, as it allows designers to predetermine the readers. For example:  
" [Policy Managers] " : "Suzy Sunshine" : "Managers"
- Make the field editable, so application Authors and Editors can select and change readers.

## **Creating a Readers field**

---

Follow these steps to create a Readers field.

Task	Procedure
1	Open a form in Lotus Domino Designer.
2	Create a field, selecting <b>Readers</b> as the field data type.
3	Specify the readers.

## **Authors Fields**

---

Users with Author access to a database may be able to create and read documents. By default, they cannot edit documents (not even the documents they create themselves). For users with Author access to edit the documents they create, they must be listed in an Authors field in the document.

This section describes how to use an Authors field to define editing capabilities for users with Author access to the database.

An **Authors field** is a type of field that works in conjunction with the Author access level in the database's Access Control List. Authors fields do not override the settings in the Access Control List. They offer developers more control over which users can edit data in the application.

Users with an access level that gives them edit privileges (Editor or higher) can edit any document that they can see, even if their names are not in the Authors field of a document. Similarly, users with a level of No Access cannot edit a document, even if their names are in the Authors field.

### **When to create an Authors field**

Include an Authors field on the form to store the name of the user when the document is created. An Authors field can store one or more user names or groups and can be editable or computed. Most Authors fields need to permit multiple values.

## How Authors and Readers Fields Work Together

The following table is a representative sample of the possible permutations of access control within a database.

ACL Level	Authors Field Exists and User Is in It	Readers Field Exists and User Is in It	Can Read Document	Can Edit Document
Author	Yes	No	Yes	Yes
Author	No	Yes	Yes	No
Author	No	No	No	No
Manager	Yes	No	Yes	Yes
Manager	No	Yes	Yes	Yes
Manager	No	No	No	No
Editor	Yes	No	Yes	No
Editor	No	Yes	Yes	Yes
Editor	No	No	No	No
Reader	Yes	No	No	Yes
Reader	No	Yes	Yes	No
Reader	No	No	No	No

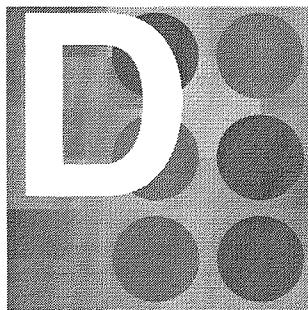
As with Readers fields, you can compute the value of an Authors field, or allow the user to select Authors for the document.

Use the @UserName function to capture the name of the current Notes user. You can use this function to dynamically compute the value of an Authors field.

## Creating an Authors field

Follow these steps to create an Authors field.

Task	Procedure
1	Create a field on the form from which the documents will be created.
2	In the <b>Field Properties</b> box, on the <b>Basics</b> tab, select <b>Authors</b> as the field type.
3	Select one of the following to determine when the field obtains its value: <ul style="list-style-type: none"><li>● <b>Editable</b> allows those who create the document to designate who can edit the document.</li><li>● <b>Computed</b> fills in the value based on a formula each time the document is refreshed.</li><li>● <b>Computed when composed</b> will evaluate only if the data does not already exist. This value is computed only once.</li><li>● <b>Computed for display</b>. Because they do not store an item or a value in a note, <b>Computed for display</b> is not appropriate for this type of field.</li></ul>
4	Enter a formula to calculate the value of the field. <b>Note:</b> Computed fields must have a formula. Editable fields can use a formula to calculate a default value that the user can change.



## **Topic D: Control Access for Anonymous Users**

After completing this topic, you should be able to:

- ✓ Define the characteristics of anonymous users.
- ✓ Identify access levels for low-security databases.
- ✓ Identify access levels for high-security databases.

### **Anonymous Users**

Users who have Internet or intranet access to a database cannot be identified by Lotus Notes in the way Notes users are identified. Lotus Domino provides ways for you to determine the maximum Internet name and password access level.

### **Setting up anonymous access**

Follow these steps to allow anonymous users to access a database.

Task	Procedure
1	Open the database in Lotus Domino Designer.
2	Open the database's Access Control List.
3	Add an entry for the group <b>Anonymous</b> .
4	Assign the access level you want to give to anonymous users.

### **Low-Security Access Control Lists**

The following table illustrates an ACL for a low-security discussion database.

Web User	Access Level
-Default-	Author
Anonymous	Author
Supervisors	Editor
Maximum Internet name and password access	Editor

## **High-Security Access Control Lists**

The following table illustrates an ACL for a high-security confidential database.

<b>Web User</b>	<b>Access Level</b>
-Default-	No Access
Anonymous	No Access
Authorized readers	Reader
Contributors	Author
Supervisors	Editor
Maximum Internet name and password access	Reader



## Practice Activity 14-3: Control Access to Documents in the Policies and Procedures Application



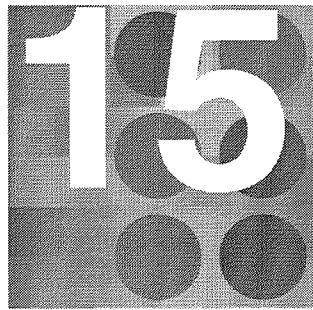
**Scenario:** In this practice activity, you will control access to documents in the Policies and Procedures application based on the following:

- All employees should be able to read any document in the application.
- The people who create documents should be able to edit them.
- Members of the Managers and Policy Makers groups should be assigned the PWrivers role.
- Only people assigned the PWrivers role should be able to create Policy documents.
- All employees can create Question documents.
- Only the person who creates a Question should be able to modify it.
- Only PWrivers should be able to create and edit Answers.



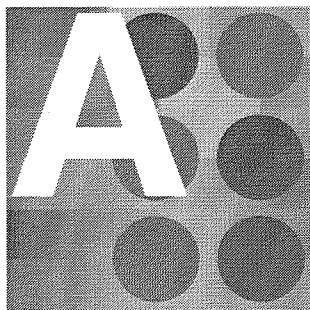
**Note:** The guided solution to this practice activity can be found in Appendix A, Solutions to Practice Activities.

1. This activity involves making changes to the ACL and the Policy, Question, and Answer forms. Determine the changes that need to be made and implement them.
2. Work with other students to help test that your application works as expected.
3. Close all open windows.



# Deploying Applications

## ■ Topic A: Deploy Applications



## **Topic A: Deploy Applications**

After completing this topic, you should be able to:

- ✓ Identify the tasks associated with deploying a Lotus Domino application.
- ✓ List the steps for updating an application's design.
- ✓ Identify the characteristics of templates.
- ✓ Identify the characteristics of and uses for master templates.
- ✓ List factors that control replication.

### **The Deployment Process**

Before deployment begins, applications are tested during development in the development environment and then previewed. However, production-level testing may also occur.

There are a number of ways to control application maintenance. The standard recommended technique uses three templates:

- A **production template**, which contains the design of the production database. You update the production database with the production template.
- A **development template**, which you use to perform updates and bug fixes. Once you finish testing the modifications, you replace the production template with the development template.
- A **backup template**, which is the previous version of the production template. You want this in case you need to roll back to the previous version. Make a backup of the production template before you replace it with the newer development template. Many sites store multiple versions of the backup template.

### **Rolling out an application**

The following checklist includes recommended steps to use when rolling out a new application.

Task	Procedure
1	Create a template out of the new database.
2	Create a new database from the newly created template.
3	Modify the ACL of the new database so that it will be properly secured and replicate correctly.