



Practice Activity 8-6: Create Reusable Code



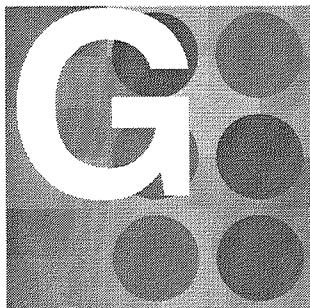
Scenario: To ensure that certain fields are populated on the Policy form in your copy of the Policies and Procedures database, create reusable code that checks to see if a value is entered in the first name, last name, and salary fields.



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. Create a reusable formula.
2. Copy it into the appropriate fields.
3. Save and test your form.



Topic G: Implement Error Checking

After completing this topic, you should be able to:

- ✓ Describe the types of run-time errors that application developers can encounter.
- ✓ Identify the characteristics of and uses for the @StatusBar function.
- ✓ Describe when the Lotus Notes client generates and reports errors.
- ✓ List @functions that can be used to handle run-time errors.

Run-Time Errors

Run-time errors are inevitable in any program. The application developer must anticipate where errors may occur and ensure that these errors do not prevent the user from working with the application.

Types of errors

The following table describes the types of run-time errors application developers encounter.

Type of Error	Description
Development	<p>Users should never see these types of errors. You should discover and fix all of them before rolling out the application. There are the following two types of development errors:</p> <ul style="list-style-type: none">● Unexpected errors are mistakes in programming. For example, forgetting a parameter in certain functions will produce the “Insufficient arguments for @function” error message at runtime, instead of compile time.● Unreported errors are incorrect results that do not report an error. For example, using the incorrect data type for a function may not produce an error message, but the function will not operate as you intended.
User	<p>These are errors that users might cause while working in the application. For example, if you ask the user for a file name and the user enters a name that does not exist. A developer cannot prevent these types of errors, but can anticipate them, test for them, and take appropriate actions to handle them so they do not cause other problems.</p>



Note: Functions that have only one signature (fixed number of required parameters) are checked at compile time. Those that have multiple signatures (the same function that operates in different ways depending on different numbers of required and optional parameters) are not.

The @StatusBar Function

To help find development errors, use @StatusBar to print data to the Lotus Notes client status bar as a basic way of debugging formulas. The status bar (at the bottom of the Lotus Notes client) stores a history of the last 20 messages to allow you to see what the formulas are doing as they execute. The following is the syntax of @StatusBar:

```
@StatusBar (message)
```

The following formula prints the result to the status bar so you can verify that the formula is functioning correctly.

```
@StatusBar ("result = " + Result);
```

Error Generation and Reporting

The Lotus Notes client **generates** an error for a field when the following occur:

- The built-in (default) validation checking fails. This occurs as soon as the user enters the value.
- The custom translation and validation functions fail. This occurs when the user attempts to save the document.

Even though errors occur, Lotus Notes only **reports** errors when the user attempts to save or refresh the document. Because Notes does not report the error until the save (or refresh), the developer can use code to correct the error or report an error with a more user-friendly message.

Error-Handling Functions

The following table describes common functions for handling run-time errors.

Function	Description
@IsError (value)	Determines if a field, variable, or expression contains an error. For example, if a user enters text in a number field.
@Return (value)	Stops the execution of a formula and returns a specified value.
@Failure (message)	Displays a message when used in a field input validation formula.
@Success	Returns 1. Use this in field input validation formulas for better readability and easier maintenance.

Examples of using @IsError

You may use @IsError to test for data entry errors. For example, you can discover if a user enters text in a number field by checking the value of the field with @IsError. If you detect an error, you can display your own message to the user, or take other action such as setting the value to zero.

This is particularly useful if you are going to perform a calculation with the number before saving the document. By using @IsError, you can avoid executing the calculation on invalid data. The following code shows an example of this:

```
NumberOfItems := @TextToNumber (Quantity);
@If (@IsError (NumberOfItems); NumberOfItems := 0; "");
```

You can also use the @IsError function to display a custom error message instead of the one Lotus Notes supplies by default. For example, if the user enters text for a number and attempts to save the document, Lotus Notes will display a message box with the **Cannot convert text to number** error. If you use @IsError in the field validation formula for the field, you can use @Prompt to override the supplied message. Below is an example of a field validation formula that displays a custom error message:

```
@If (@IsError(@ThisValue));
  @Failure ("Please enter a number for " + @ThisName + " .");
  @Success)
```



Practice Activity 8-7: Implement Error Interception

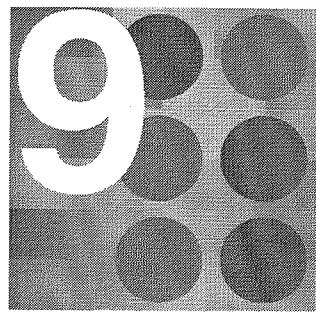


Scenario: The Worldwide Corporation frequently updates their company policies and would like to allow for internal review of any modifications. Revise the Policies and Procedures database to allow for a one month review period before the policy is implemented, and to intercept run-time errors before they need to be handled.



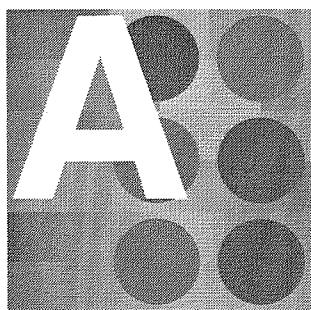
Note: The guided solution to this activity can be found in Appendix A, Solutions to Practice Activities.

1. In the **Policies and Procedures** database, add a row called **Modifications to the Policy** to the existing table.
2. Create a Computed Date/Time field called **L_ModifiedDates**
Specify the following:
 - Allow multiple values
 - Display date only
 - Display separate values with new line
 - Hide paragraph if the field is empty
3. The field needs to be modified only when the document is actually saved. Write a formula that determines if the field needs to be modified. Have the formula check if/when the document has been saved, and add today's date to the end of the list of values if the document is being saved again.
4. On the line below the new field, but in the same table cell, enter "N/A" and hide it when **L_ModifiedDates=NULL**.
5. Revise the formula for the **Effective Date** field to use the **L_ModifiedDates** field.
6. Test your form.
7. Close all open windows.



Using Formulas in Views

- **Topic A:** Create Selection Formulas
- **Topic B:** Concatenate Values in a View Column
- **Topic C:** Display Icons In A View



Topic A: Create Selection Formulas

After completing this topic, you should be able to:

- ✓ Describe view events.
- ✓ Identify the characteristics of and uses for view selection formulas.

View Events

View events occur when users work in a specific view or folder. View events include opening or closing a view, adding documents to a calendar view, or adding documents to a folder. Examples of view and folder events are described in the following table.

Event Name	Result
ViewSelection	Selects the documents that the view displays.
QueryOpen	Prevents users from opening a view in certain circumstances such as from opening a "month end" view before the twentieth day of the month.
PostOpen	Creates a new document or open an existing one.
QueryRecalc	Informs users before a large view refreshes that it could take a while and ask them if they really want to proceed.
QueryPaste	Prevents users from pasting documents into the database.
PostPaste	Changes the value of StartDateTime and EndDateTime in the document when you paste an appointment on a particular day and time slot.
QueryClose	Prevents users from closing a view such as when there are still action items in the action item view assigned to them.

Lesson 9 ■ Using Formulas in Views

Event Name	Result
QueryAddToFolder	Lets you prevent someone from dragging a document from one folder or view to another. For example, you can prevent documents from moving to the Done folder if their status is still Open. Note that the trigger for this event is in the view or folder from which the document is moved.

In addition to the events available for all views, calendar views have specific events. Examples of calendar view events are shown in the following table.

Event Name	Result
RegionDoubleClick	Creates a new document when users click an area in a calendar view.
QueryDragDrop	Prevents someone from dropping an appointment on an inappropriate day or time, such as a weekend day or a time that is after 5:00 PM.
PostDragDrop	Changes the value of StartDateTime and EndDateTime in the document after you drop an appointment on a particular day and time slot.

View Selection Formulas

Views can include all documents in a database or a subset of those documents. The selection condition, whether it is created with Search Builder or a view selection formula, controls the document displayed. All documents that match the selection criteria are displayed in the view.

Earlier in the course, you used simple conditions to select which documents to display in a view. View selection formulas, however, provide more control over which documents a view displays.

Including documents in a view

The following figure illustrates the view selection process.

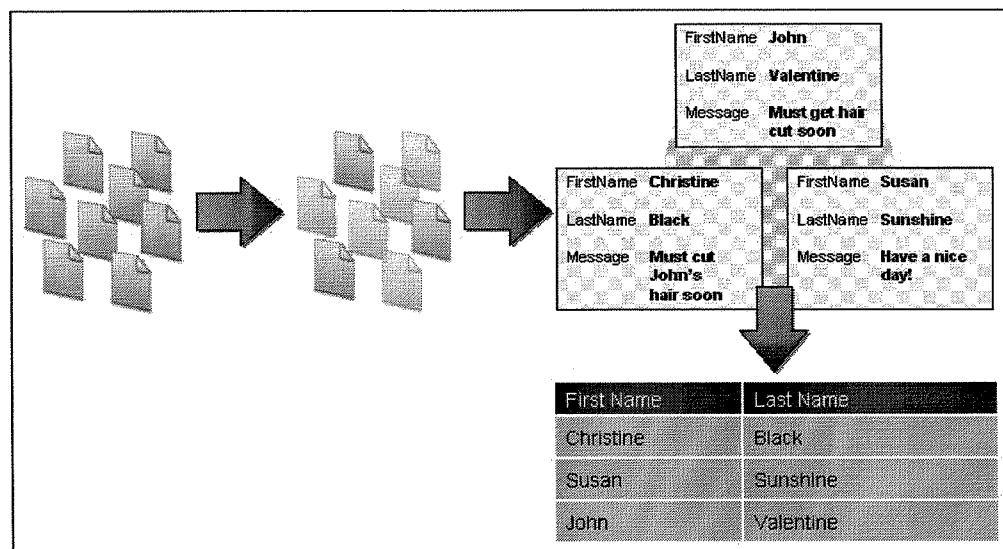


Figure 9-1: The View selection process

The view selection formula

When a view is first created, the default selection formula is `SELECT @All`. Using this formula, all documents in the database are included in the view. The `SELECT` statement allows greater control of the selection of documents.

The view selection formula syntax is as follows:

```
SELECT Condition formula;
```

The following table describes the parts of a view selection formula.

Part	Function
SELECT keyword	Includes a document based on the result of the comparison formula.

Part	Function
Condition formula	A formula that evaluates to either True (1) or False (0) for each document in the database. It can use constants, fields, operators, and @functions.

Examples of view selection formulas

The following table describes a few examples of view selection formulas.

This View Selection Formula...	Includes All Documents That...
SELECT @All	Are in the database. This is the default formula for view selection.
SELECT Form = "ei"	Were created or last modified using a form named ei.
SELECT (Price < 1000) & (InStock = "Yes")	Contain a field called Price whose value is less than 1000 and a field called Instock whose value is Yes.
SELECT Form = "ei" & Department != "Accounting"	Were created or last modified using the form named ei and do not contain a field named Department whose value equals Accounting.



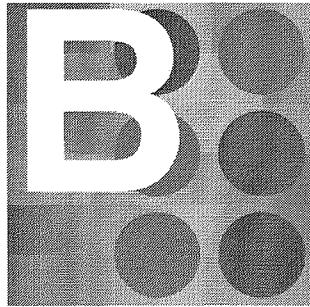
Activity 9-1: Write a view selection formula



Scenario: In this activity, you will create a view selection formula that selects documents of employees whose salaries are over \$30,000.

Follow these steps to write a view selection formula.

Step	Action
1	Open your copy of the Practice database in the Lotus Domino Designer. Click Views in the Design Elements list. Result: A list of views appears in the Work pane.
2	Create a view based on the Employee Information view. Name the view Salaries Result: The view is created and appears in the Design list.
3	Open the view. Result: The view appears in the Work pane.
4	Select View Selection in the Objects pane.
5	In the Programmer's pane, select Formula in the Run drop-down list. Result: <code>SELECT ((Form = "Employee Information") (Form = "ei"))</code> appears in the script area
6	Delete the selection formula. Enter the following formula: <code>SELECT Form="ei" & Salary>30000</code> Result: The new formula is created.
7	Save the view. Preview your work in the IBM® Lotus Notes® client.



Topic B: Concatenate Values in a View Column

After completing this topic, you should be able to:

- ✓ Identify the characteristics of and uses for concatenation.
- ✓ Identify ways to convert values to text in a view column.
- ✓ Identify the characteristics of and uses for the @Text function.
- ✓ Concatenate multiple values in a view column.

Concatenation

Concatenating text values, or strings, is useful when you want to combine related fields in a single column or when the field value requires supporting text to make sense to users.

The + sign is used to concatenate text into a single string. For example, the following view column formula displays the first name and last name with a space between them in a single column.

```
FirstName + " " + LastName
```

You can only concatenate text values. To include other data types in your concatenation, do the following:

- Convert (also known as normalize) the values to text.
- Concatenate the text values.

Converting Values To Text

To combine different types of values, you must first make sure that the values are the same data type. For example, to display number and text values in a view column, convert the number to text, then concatenate the text values.

Use @Text to convert number and date-time values to text. @Text offers options to control the format of the resulting text.

The syntax for @Text is as follows:

```
@Text ( value ; "format-string" );
```

Where:

- Value can be a number, time-date, text, or rich text.
- Format-string is optional. It controls the formatting of the returned text value. For example, if a currency value is converted to text, you might want the new value displayed with the appropriate currency symbols and punctuation.

The @Text Function

There are two types of format string components available for text values: datetime components and numeric value components. These components can be combined in various ways. Listed here are several of the more commonly used string components and what they display:

- D0 – year, month, day
- D2 – month and day
- S1 – Time only
- C – Currency

The following table provides some examples of converting values to text.

Original Value	Formula	Result
The Sales field contains the value 800.	@Text (Sales; "C,2")	\$800.00
The Date field contains 12/31/2005 11:59:59 PM.	@Text (Date; "S0D2")	12/31
The Date field contains 12/31/2005 11:59:59 PM.	@Text (Date; "S1T0")	11:59:59 PM.

Writing a formula that concatenates multiple values in a view column

Follow these steps to enter a column formula that concatenates multiple values.

Task	Procedure
1	In the Object list, select the Column Value property under the view.
2	Write a formula to concatenate the values: <ul style="list-style-type: none">● Use functions to convert number and date-time values to text.● Combine the converted values using the concatenation operator (+).
3	Save and preview the view.



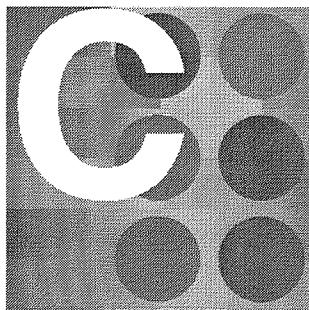
Activity 9-2: Concatenate text and number values



Scenario: In this activity, you will create a view that quickly shows a list of employees and their date of hire. You will edit your Employee Information form to include an editable Date Hired field, with a type of Date/Time. You will then edit the documents in your Employees view, and give them a hiring date, and leave one of them blank.

Follow these steps to create a Date Hired field, assign employee hire dates, and concatenate text and number values.

Step	Action
1	In your copy of the Practice database, add a field named Date Hired to your Employee Information form. Give it a type of Date/Time and make it editable .
2	Using your Employee Information view, open all but one of the documents, and give the employees a date of hire.
3	Create a view in the Practice database called Date of Hire . Result: The view appears in the Work pane.
4	Open the new view and select the default column. Delete the column header. Result: The default column header is deleted.
5	In the Script area for the column, select Formula .
6	Replace the default formula with: <code>FirstName + " " + LastName + " was hired on " + @Text (DateHired; "S0D1")</code> ; Result: Employees' name and hire dates will appear. If an employee was hired this year, only the day and month will appear.
7	Sort the column in ascending order.
8	In the View Selection object, select the formula, and type the following formula: <code>SELECT Form= "ei"</code> Result: The view will only display documents created with the Employee Information form.
9	Save and preview the view.



Topic C: Display Icons In A View

After completing this topic, you should be able to:

- ✓ Display icons in a view.
- ✓ Describe the requirements for creating a column icon formula.
- ✓ Add a predefined icon to a column.
- ✓ Add a custom icon to a view column.

Displaying Icons In A View

One way to make different documents stand out in a view is to associate an icon with the document. In order to display icon columns:

- The column property **Display values as icons** must be set.
- A column formula must be created which evaluates to a number corresponding to the preset list of IBM® Lotus® Domino® icons, or that evaluates to the name (or alias) of a custom icon stored as a Shared Image Resource.

Column properties

Select the **Display values as icons** option in the Column Info tab of the Column Properties box. The following figure shows this option selected.

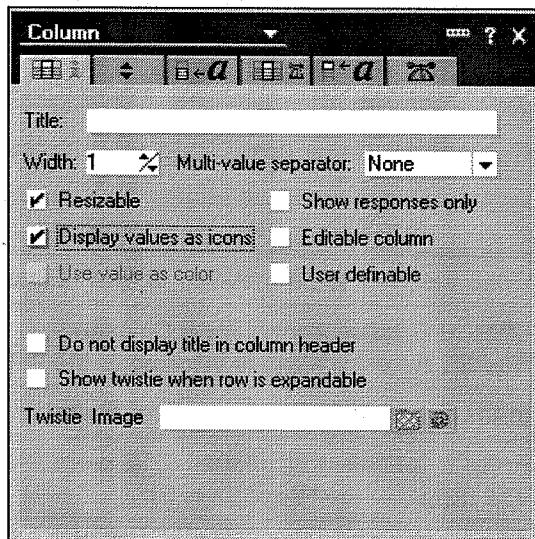


Figure 9-2: The **Display values as icons** option

Creating A Column Icon Formula

In order for a view column to display an icon a formula must be created that evaluates to the icon. Typically, the formula evaluates to a number that represents a pre-defined icon, although it can evaluate to a custom icon.

The following formula displays a thumbs up icon next to products that have been approved:

```
@If (Approved="Yes";83;0)
```

If the approved field contains anything other than the word yes, then nothing is displayed.

Predefined icon set

Figure 9-3 displays the predefined set of icons available in Lotus Domino.

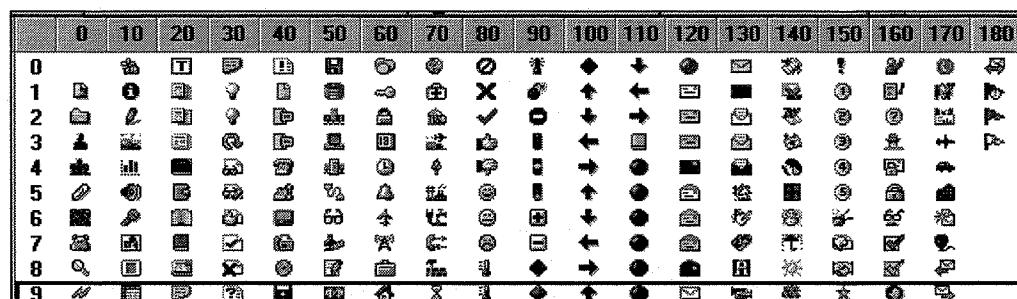


Figure 9-3: A predefined set of icons available in Lotus Domino.

Adding a predefined icon to a column

Follow these steps to add a predefined icon to a view column.

Task	Procedure
1	Open the view to be modified in IBM® Lotus® Domino Designer®.
2	Choose Create→Insert New Column . Result: A new column is inserted to the left of the selected column.
3	In the Programmer's pane, select Formula .
4	Enter a formula that results in a predefined icon value. For example: <code>@If (Approved="Yes";83;0)</code>
5	Choose Design→Column Properties . Result: The Column Properties box opens.
6	Select Display values as icons .
7	Save and preview the view.



Activity 9-3: Designate multiple icons for a specific type of document



Scenario: In this activity, you will use the Sample Employee Information view in your copy of the Practice database. Write a formula that uses the hire date to determine those employees who are fully vested in the company retirement program. Create a smiley icon for those fully-vested employees hired before 2000, and a checkmark icon for those employees hired after 1/1/2000.

Follow these steps to add multiple icons to your Sample view.

Step	Action
1	Open the Employee Information view in your copy of the Practice database.
2	Add a new column to the left of the first column.
3	Open the Column Properties box.
4	On the Column Info tab, reduce the width to 1 and select Display values as icons .
5	In the Programmer's pane, select Formula , and type the following formula: <code>@If(DateHired<[01/01/2000];85;82)</code> Note: The number zero indicates that no icon will be displayed. Result: Those who were hired before 01/01/2000 have a smiley face, those who were hired on 01/01/2000 or later have a checkmark. Anyone with a blank hiring date will have no icon because the inequality evaluates to an error condition, which stops all processing for that column in that row.
6	Press Shift+F9 to rebuild the view.
7	Close all open windows.

Adding A Custom Icon To A View Column

To use a custom icon in a view column, the image must be saved as an image resource. The image can be in one of the three graphic formats:

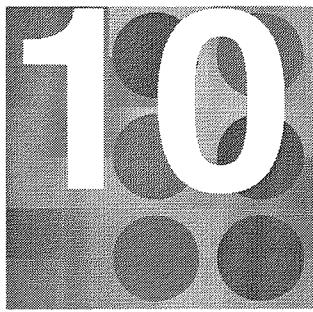
- GIF
- BMP
- JPG

The recommended size for a custom column icon is 17x17 pixels. You cannot combine a pre-defined and a custom icon in the same column.

Adding a custom icon to a view column

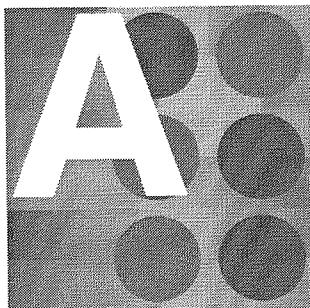
Follow these steps to add a custom icon to a view column.

Task	Procedure
1	Open the database in Lotus Domino Designer.
2	Select Shared Resources → Images from the Design Elements list.
3	Click New Image Resource .
4	Select the image to be added and click Open . Result: The image is added to the Work Pane list. For example: logo.jpg.
5	Open the view to be modified in Lotus Domino Designer.
6	Choose Create → Insert New → Column . Result: A new column is inserted to the left of the selected column.
7	In the Programmer's pane, select Formula .
8	Enter a formula to display the shared image. For example: <code>@IF (Approved="Yes"; "logo.jpg"; 0)</code>
9	Choose Design → Column Properties . Result: The Column Properties box opens.
10	Select Display values as icons .
11	Save and preview the view.



Enhancing Forms

- **Topic A:** Work with Layers
- **Topic B:** Work with Subforms
- **Topic C:** Create Shared Fields
- **Topic D:** Enable Inheritance in Forms and Documents
- **Topic E:** Create Related Forms
- **Topic F:** Enable AutoSave
- **Topic G:** Implement Lotus Sametime Instant Messaging in Forms



Topic A: Work with Layers

After completing this topic, you should be able to:

- ✓ Identify the characteristics of and uses for layers.
- ✓ Create a layer on a form.
- ✓ Describe layer anchors.
- ✓ List properties associated with layers.
- ✓ Modify the background of a layer.
- ✓ Describe ways to resize and reposition layers.
- ✓ Describe layer position values.
- ✓ Set a layer's coordinates.
- ✓ Describe the different methods for aligning multiple layers on a form.
- ✓ Hide a layer.
- ✓ Work with layers.

Layers

Layers are used to position overlapping blocks of content on a form, subform, or page. This provides you with more control over the design of IBM® Lotus® Domino® pages and forms. Make note of the following when working with layers on forms and pages:

- Layers on a form or subform can contain anything a form can contain.
- Layers on a page can contain anything a page can contain.

This topic focuses on using layers on a form.

Benefits of working with layers

There are many benefits to working with layers. Benefits include:

- Precise positioning to allow for complex layout options.
- Stacking layers provides great design control and flexibility. For example, different layers containing text and images can be stacked for a high-impact page title.
- Layers can be transparent or opaque, depending upon the type of effect desired.

Content of a layer

Add content to a layer just as you would add content to a form. You can copy and paste from another form or layer, or type directly in the layer. You can also add graphics.

Creating a layer

Follow these steps to create a layer on a form.

Task	Procedure
1	Open a form, page, or subform in IBM® Lotus® Domino Designer®.
2	Place the cursor where the layer should appear.
3	Choose Create→Layer . Result: A blank layer is inserted onto the form. Selection handles surround the layer, indicating it is selected.
4	Click inside the layer and add the layer's content.
5	Save and preview the form.

Layer Anchors

Each layer has an anchor. The layer anchor associates itself with the element closest to where it was inserted, such as a paragraph of text. This is referred to as the parent element.

If the layer is moved, the anchor remains in place. To display or change the name of the layer, right-click the layer anchor. The Layer Anchor Properties box appears with the current name of the layer.

The following figure shows a layer and its anchor.

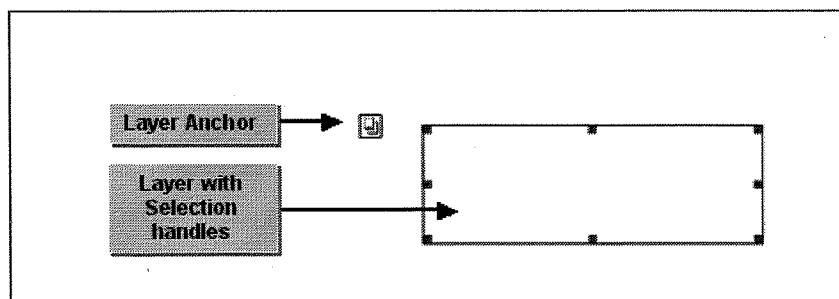


Figure 10-1: A layer anchor

Layer Properties

After you create a layer, you can control the following properties:

- Background color and image
- HTML properties
- Position

Layer Background Color

You can set the layer's background color to be the system color, gray, transparent, black, or an RGB value.

The following table lists the color icon choices on the Background tab of the Layer Properties box.

Icon	Function
	System color
	Transparent
	Black (0,0,0)
	Red, Green, Blue (RGB)

Layer Background Images

Layers can contain a background image in addition to, or instead of, a background color. The image used for the layer background must be a shared image resource. Images can be displayed once in the layer, or tiled horizontally, vertically, or continuously.

Modifying the background of a layer

Follow these steps to modify a layer's background.

Task	Procedure
1	Select the layer.
2	Select Layer→Layer Properties to open the Layer Properties box.
3	Click the Background tab
4	To change the layer's color, click the drop-down arrow to the right of Color .

Task	Procedure
5	Choose one of the color options: <ul style="list-style-type: none">● System● Transparent● Black● RGB
6	Click on the folder, to the right of Image Source , to insert an image resource.
7	Select one of the image repeat values.
8	Save and preview the form.

Layer Size and Position

A layer can be resized and repositioned several different ways. You can:

- Drag and drop it.
- Explicitly set its position.
- Position it relative to other layers on the form.

Layer Position Values

The position of a layer on the screen, its size, and the position of the layer in relationship to other layers is displayed, modified, or both, in the Layer position tab of the Layer Properties box.

When you drag and drop a layer, or resize it using the layer window handles, the Top, Left, Width, and Height parameters change to reflect the new position or size. These values can also be changed to set the size and position of a layer explicitly.

The **Z-Index** parameter determines the order in which the layer appears if there are multiple layers on the form, page, or subform. Think of the layers as stacked one upon the other. If there were three layers, the lowest layer would have a Z-Index of zero, the middle layer would have a Z-Index of one, and the top layer would have a Z-Index of two. Other facts about the Z-Index parameter:

- 0 is the default.
- 1 appears above 0.
- Multiple layers can have the same level.
- Negative values are placed behind the parent element.

Setting a layer's coordinates

Follow these steps to modify the layer's position properties.

Task	Procedure
1	Select the layer.
2	Choose Layer→Layer Properties .
3	Click the Positioning tab.
4	Enter the appropriate coordinates to align the layer from the top or left edge of the parent element.
5	Enter the appropriate coordinates to change the layer's height and width.
6	Enter the appropriate integer for the layer's Z-Index.
7	Save and preview the form.

Multiple Layer Alignment

There are different ways to align multiple layers. The following table lists the methods for aligning multiple layers.

To	Follow These Steps
Select multiple layers.	<ul style="list-style-type: none">● Select the first layer.● Press CTRL and click the other layer(s). <p>Result: Selection handles appear around the selected layers.</p>
Align multiple layers.	<ul style="list-style-type: none">● Select the layers to align.● Choose Layers→Align.● Select from the available options.
Adjust layer sizes.	<ul style="list-style-type: none">● Select the layers to adjust.● Choose Layer→Make same size.● Select from the available options.

Hiding Layers

Occasionally, you will want to hide a layer temporarily to make layout adjustments.

Like other design elements, layers can be hidden using Hide/When properties. However, for ease of development Lotus Domino Designer provides another method for hiding layers. That method is the Layer Tree.

The Layer Tree

The **Layer Tree** hides one or more layers for the current session. You can work on one layer without another layer, or layers, getting in the way. If a layer is hidden, a grey circle appears next to the name of the layer; otherwise, the circle is empty. The following figure shows a Layer Tree Properties box with three hidden layers.

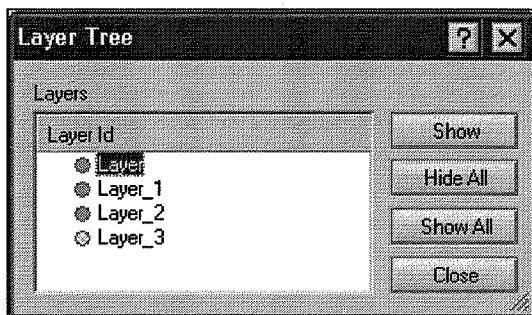


Figure 10-2: The Layer Tree



Note: When a hidden layer is selected, the top button in the Properties box toggles from **Hide** to **Show**.

Hiding a layer

Follow these steps to hide a layer.

Task	Procedure
1	Choose Design→Layer Tree .
2	Select the name of the layer you want to hide.
3	Click Hide . Result: The layer will not be visible for the current session.



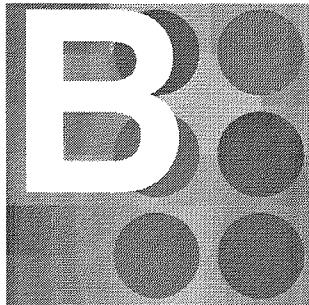
Activity 10-1: Work with layers



Scenario: Layers can be used to enhance the appearance of the Employee Information form you created in an earlier activity. In this activity, you will add a layer with a background image for visual appeal. You will copy the form fields and labels into the layer.

Follow these steps to create a layer.

Step	Action
1	Open the Employee Information form in your copy of the Practice database. Result: The form appears in the Work pane.
2	Place the cursor below the other content on the form.
3	Choose Create→Layer . Result: A blank layer is added to the form.
4	Drag the layer handles so that the layer is approximately as large as the area of the field labels and fields on the form.
5	Highlight the fields and the label text. Press Ctrl+X to cut the fields.
6	Place the cursor inside the layer. Press Ctrl+V to copy the fields into the layer. Result: The fields appear inside the layer.
7	Resize and reposition the layer by dragging it so that it is under the form title. Result: The layer, along with its fields and text, is repositioned.
8	Choose Layer→Layer Properties . Result: The Layer Properties box appears.
9	Click the Background tab.
10	Click the folder icon next to the Image Source text box. Select lesson_menu_background.gif . Click OK . Result: The background image appears in the layer.
11	Select Repeat horizontally in the Repeat drop-down list. Result: The background image fills the layer.
12	Select the HTML tab.
13	Type Image_and_Fields in the Id field.
14	Save and preview the form.



Topic B: Work with Subforms

After completing this topic, you should be able to:

- ✓ Describe subforms.
- ✓ Describe how subforms work.
- ✓ Create a subform.
- ✓ List properties associated with subforms.
- ✓ Insert a subform in a form.
- ✓ Identify the characteristics of and uses for computed subforms.
- ✓ Implement computed subforms
- ✓ Insert computed subforms in forms.

Subforms

A **subform** is a collection of form elements stored as a separate design element. A subform can contain anything a form can contain.

A form can contain one or more subforms. One subform can be used in many different forms within the same database or across different databases. Subforms can also be used conditionally, depending on the results of a formula. Subforms can also contain other subforms.

The benefits of using subforms in your application include:

- Saving design time when the same set of design elements need to be included in more than one form
- Ensuring design consistency among the forms in the application

The following figure shows one subform being used in multiple forms.

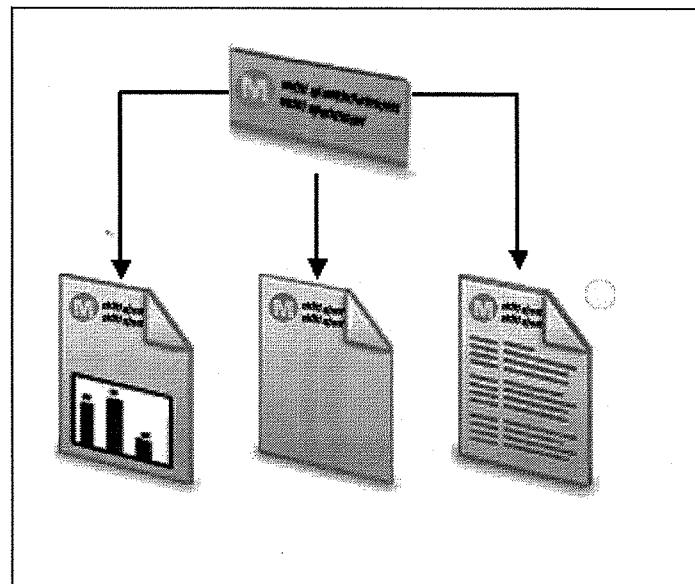


Figure 10-3: Subforms

When to use subforms

Examples of when to use a subform include:

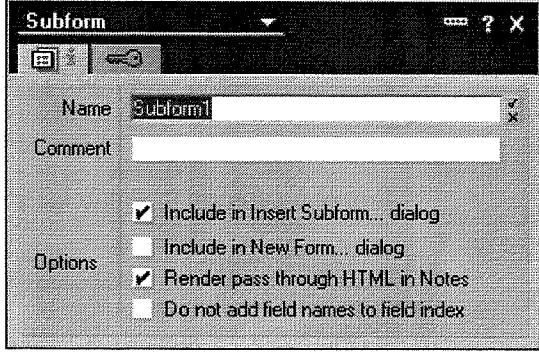
- To add a formatted logo or letterhead
- To ensure consistency and standards in form design
- To start building new forms quickly

How Subforms Work

Subforms can only be used as part of a form. Every time a document is created or opened, the form references the subform. Elements of a subform automatically become part of the form when displaying documents. Changes made to a subform affect all forms and documents that include the subform.

Creating a subform

Creating and using a subform in a database is a two-part process. First, the subform needs to be created. Then the subform needs to be inserted as part of a form. Follow these steps to create a subform.

Task	Procedure
1	Open the database in Lotus Domino Designer.
2	Choose Shared Code in the Design list.
3	Select Subforms from the Shared Code list.
4	Click New Subform in the Work pane. Result: A new blank subform opens in the Work pane.
5	Choose Design→Subform Properties to open the Subform Properties box. Note: The Include in Insert Subform... dialog option is selected by default. 
6	Name the new subform.
7	Add elements to the subform as you would a form.
8	Save and close the subform.

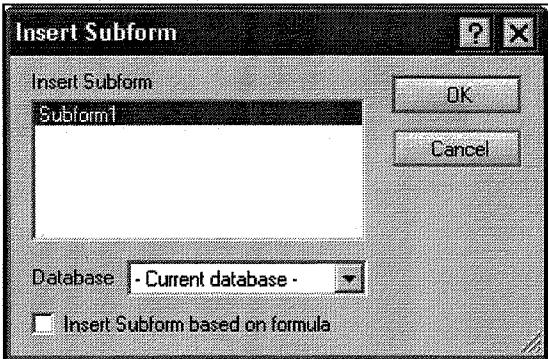
Subform Properties

Like other design elements, subforms have properties you can set, using the Subform Properties box. Use the Subform Properties box to:

- Give the subform a name and an alias.
- Specify display options for the subform.

Inserting a subform in a form

Follow these steps to insert a subform in a form.

Task	Procedure
1	Open a form in Lotus Domino Designer.
2	In the Work pane, click where the subform will appear.
3	Choose Create→Resource→Insert Subform . Result: The Insert Subform dialog box appears. 
4	Select a subform from the list that appears.
5	Click OK . Result: The subform appears in the specified place in the form.

Computed Subforms

You can have subforms appear conditionally, depending on the result of a formula. This is called a computed subform. For example, you might offer users a choice of custom mail forms with different graphics and styles for various types of messages.

Implementing computed subforms

Follow these steps to implement computed subforms.

Task	Procedure
1	Create the subform(s).
2	Decide under what conditions each subform will appear.
3	Write a formula to include the subform(s) based on those conditions.
4	Add the formula to the appropriate form(s).

Creating a computed subform formula

You could use the following formula to select and insert a subform based on whether a user is posing a question, replying to a question, or adding a comment.

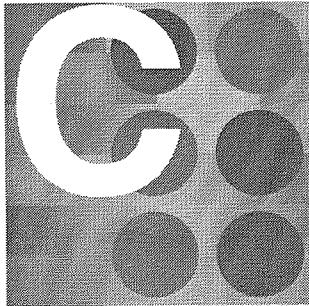
```
@If (MsgType="question"; "QuestionSub"; MsgType="reply"; "ReplySub";  
"CommentSub")
```

The value in the MsgType field determines which subform is used.

Inserting computed subforms in forms

Follow these steps to insert a computed subform in a form.

Task	Procedure
1	Open the form.
2	Click where you want to add the subform.
3	Choose Create→Resource→Insert Subform .
4	Select Insert Subform based on formula .
5	Click OK .
6	Enter a formula in the Programmer's pane that determines which subform to display.
7	Save the form.



Topic C: Create Shared Fields

After completing this topic, you should be able to:

- ✓ Describe shared fields.
- ✓ Create shared fields.
- ✓ Insert shared fields.

Shared Fields

You can define a field for use on more than one form. For example, many forms have a creation date field; you can define this field once and reuse it. When you define a field as a shared field, Lotus Domino Designer displays the field with a dark border and adds the field name to a list of shared fields available for use in a database.



Note: In multiple-database applications, you can use shared fields from separate databases.

Benefits of Shared Fields

The benefits of using shared fields include:

- Maintaining and reusing consistent formulas throughout an application
- Providing single point maintenance for a given field
- Standardizing keyword lists

Creating a shared field

Follow these steps to create a shared field.

Task	Procedure
1	Expand Shared Code in the Design pane. Click Fields .
2	Click New Shared Field .
3	Enter a name for the shared field.
4	Assign a field type and choose Editable or a computed option, if applicable.
5	Close the Properties box.
6	Close and save the shared field.

Inserting shared fields

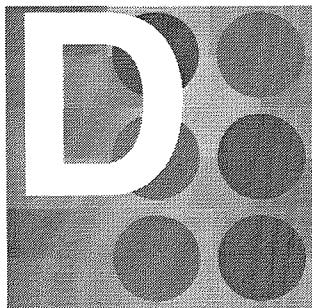
Follow these steps to insert a shared field onto a form or subform.

Task	Procedure
1	Open a form or subform and position the cursor where you want the field to appear.
2	Choose Create→Resource→Insert Shared Field . Result: The Insert Shared Field dialog box appears.
3	Select the shared field you want to use and click OK . To select a shared field from another database, click the Database pull-down list, highlight a database, select a shared field in that database, and click OK .

Converting single to shared fields

Follow these steps to convert a single-use field to a shared field.

Task	Procedure
1	Open the form.
2	Click the field to be shared.
3	Choose Design→Share This Field .



Topic D: Enable Inheritance in Forms and Documents

After completing this topic, you should be able to:

- ✓ Identify the characteristics of and uses for inheritance.
- ✓ Describe requirements for enabling inheritance.
- ✓ Enable inheritance on a form.

Inheritance

Lotus Domino Designer allows you to create fields that inherit values from other fields. The source fields and target fields can be in separate documents or in the same document. Therefore, data can be entered or assigned once and reused as needed in multiple places.

For example, when an employee is first hired, there are many forms that need to be completed that include the employee's name, address, employee number, job title, and so on. Using inheritance, this information need only be entered once.

Lotus Domino supports this sharing of information by providing a method for inheriting information from one document to another.

Benefits of inheritance

The ability to inherit information automatically from one document to another has a number of benefits, including:

- It is more efficient than typing the same information multiple times.
- It helps preserve the accuracy of the information contained in each document.

How inheritance works

Fields can inherit values from fields in other documents in the same database or from fields on the same document. The following diagram illustrates how one source document contains field data that can be inherited by multiple receiving documents.

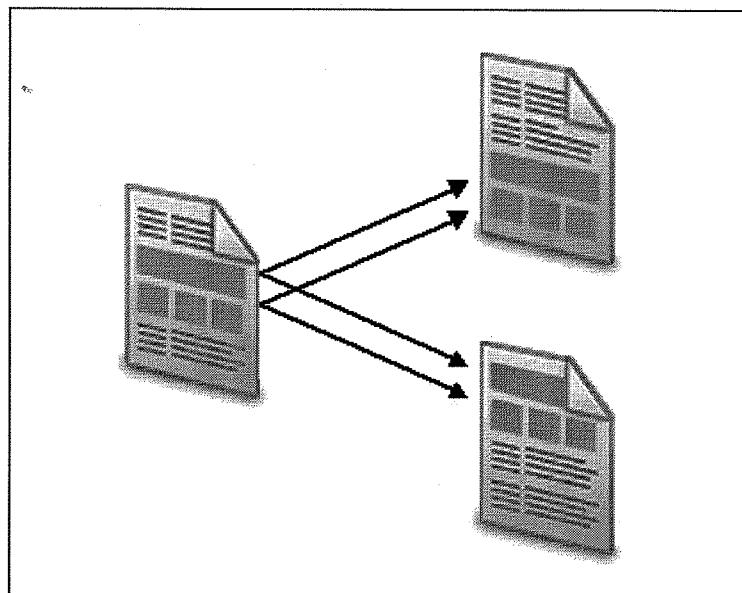


Figure 10-4: Inheritance

Enabling Inheritance

Forms can be designed to inherit information from an existing document. Creating a form to inherit values from another document requires that the developer:

- Enable the **On Create: Formulas inherit values from selected document** form property.
- Create field(s) containing the following characteristics:
 - **Computed when composed** field type
 - Includes a formula that specifies the source field

Enabling inheritance on a form

Follow these steps to enable inheritance for a form.

Task	Procedure
1	Create a form or open an existing form in Lotus Domino Designer.
2	Open the Form Properties box. On the Defaults tab, select On Create: Formulas inherit values from selected document .
3	Specify a computed field type for the field(s) that will inherit information.
4	Write a formula for the inheriting field(s) that evaluates to the source field name.
5	Save the form.
6	To test your work, highlight an existing document that contains the field(s) to inherit, and then create a document with the new or revised form.  Tip: Fields that inherit information do not need the same field name as their source fields; they do, however, need to be the same data type. For consistency sake, an inherited field's name should be kept the same as the original field if it contains the same information. This allows for more consistent programming methods and view formulas.



Activity 10-2: Enable inheritance



Scenario: In this activity, you will create an Employee Benefits form that inherits an employee's first and last name from documents created earlier in the course.

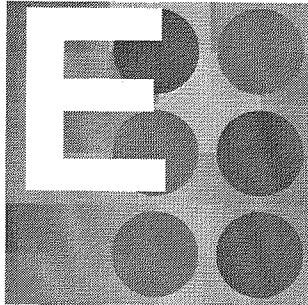
Follow these steps to enable inheritance.

Step	Action
1	In your copy of the Practice database, create a form and name it Employee Benefits with an alias of eb Result: The form appears in the Work pane.
2	Choose Design→Form Properties. Click the Defaults tab. Result: The Defaults tab of the Form Properties box appears.
3	Select On Create: Formulas inherit values from selected document. Close the Properties box. Result: The property is set and the Form Properties box closes.
4	Create a field with the following characteristics: <ul style="list-style-type: none"> ● Name: InheritFName ● Type: Text, Computed when Composed Result: The InheritFName field is created.
5	In the Programmer's pane for the InheritFName field, enter the formula: FirstName Result: The FirstName field is added as a formula.
6	Create a field with the following characteristics: <ul style="list-style-type: none"> ● Field name: InheritLName ● Type: Text, Computed when Composed ● Formula: LastName Result: The InheritLName field is created.
7	Create a rich text field named Comments.
8	Save your changes.
9	Create an Employee Benefits view in order to see the new documents you will create.
10	In the Lotus Notes client, open the Employees view. Select one of the employee documents that you created earlier in the course.
11	Choose Create. Select Employee Benefits. Result: A new Employee Benefits form opens with the appropriate fields populated from the selected document.
12	Save the new document.

Topic D: Enable Inheritance in Forms and Documents

Lesson 10 ■ Enhancing Forms

Step	Action
13	Verify that the new document is displayed properly, and that it displays the correct content.



Topic E: Create Related Forms

After completing this topic, you should be able to:

- ✓ Describe related forms.
- ✓ Create a response form.
- ✓ Describe the relationship between main documents and response documents.
- ✓ Describe the relationship between inheritance and related documents.
- ✓ Describe threaded discussions.

Related Forms

In almost every application there is a need to create information that is related to other information. A typical example of this is a question and answer structure. Another very familiar example is a discussion forum.

Form types

Lotus Domino provides for a main document/response document structure, sometimes referred to as a response hierarchy, by letting the developer designate a form's type. The following table lists the three Lotus Domino form types and how they are used.

Form Type	Description
Document	This document type is the default for Lotus Domino forms. It is used to create the top level in a hierarchy of documents. These documents are sometimes referred to as main or parent documents.
Response	This document type is a child to the Document type. It is referred to as a response document.
Response to response	This document type is a child to the Response document type. It is referred to as a response to response document.

The response hierarchy

The following figure illustrates the relationship between main documents, response documents, and response to response documents.

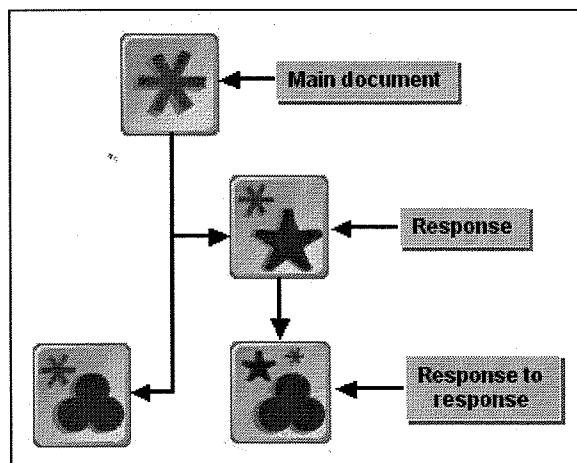


Figure 10-5: Response hierarchy

Designating a Form as a Response

By default, the form type is set as Document. To designate that a form is a response or a response to a response, change the Form Type property. The following figure shows the Form Type property.

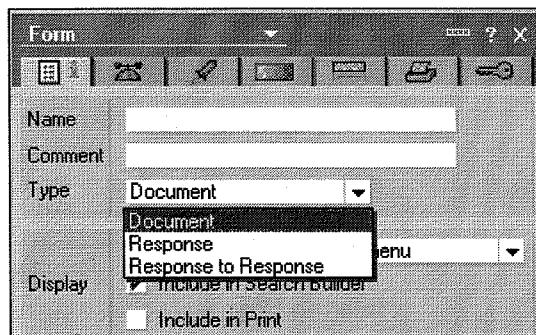


Figure 10-6: Form Type properties

Creating a response form

Follow these steps to create a response form.

Task	Procedure
1	Create a form or open an existing form in Lotus Domino Designer.
2	Choose Design→Form Properties .
3	On the Form Info tab, select one of the following: <ul style="list-style-type: none">● Response for the form to create responses to main documents.● Response to Response for the form to create responses to other responses.
4	Save the form.

Document Relationships

To properly establish the relationship between a main document and a response document, you must have the main document selected in a view or opened when the response is created.

Inheritance and Related Documents

Inheritance is frequently used to transfer information between main and response documents. For instance, in a discussion application you typically repeat the subject of the main document in the response. As with document to document inheritance, the main document must be selected or open when a response is created for inheritance to work correctly.

Threaded Discussions

When developers design any type of main document/response document application, sometimes referred to as a threaded discussion, it is good practice to consider how users keep track of the information in the main document. Using Lotus Domino form properties, the developer can design the response form to:

- Create a link to the main document in a rich text field.
- Copy the contents of the main document as collapsible rich text.
- Copy the contents of the main document into a rich text field.

Linking to the main document takes up much less disk space and is generally the preferred method.



Activity 10-3: Create related forms



Scenario: In this activity, you will make the Employee Benefits form a response form and include a link to the parent document.

Follow these steps to create related forms.

Step	Action
1	<p>Open your copy of the Employee Benefits form in Lotus Domino Designer.</p> <p>Result: The form appears in the Work pane.</p>
2	<p>Create a field with the following characteristics:</p> <ul style="list-style-type: none"> ● Name: ParentDoc ● Field type: Rich Text and Computed <p>Result: The new field is created.</p>
3	<p>Choose Design→Form Properties.</p> <p>Result: The Form Properties box opens.</p>
4	<p>Click the Defaults tab. Select On Create: Inherit entire selected document into rich text field.</p> <p>Result: You will see the choices shown in the following graphic.</p>
5	<p>Select the ParentDoc field from the drop-down list. Select Link.</p> <p>Result: The form properties are set.</p>
6	<p>On the Form Info tab, select Response as the form type.</p>
7	<p>Save the form.</p>
8	<p>To test your changes, select a document in the Employees view and create an Employee Benefits document. Note that there is now a link to the parent document.</p>



Practice Activity 10-4: Create Question and Answer Forms in the Policies and Procedures Database



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



Scenario: The Worldwide Corporation wants to make it easier for employees to ask questions and receive answers about company policies. In this activity, you will create Question and Answer forms. The Answer form needs to inherit information from the Question form.

Both forms should have a consistent look and feel.

This activity has the following parts:

- Use a layer and the mod_menu_background.gif file to enhance the look of the forms.
- Create a subform to provide a consistent header to both forms.
- Design and create the Question form.
- Design and create the Answer form.

Later in the course, you will create a view to display the question and answer documents correctly.

1. Guidelines for the Question Form

The Question form needs to gather certain information. The following table describes the information to be gathered and lists suggested field names and types.

Information to Be Gathered	Suggested Field Name and Type
The user's first name	FName, Text
The user's last name	LName, Text
The category that the question falls into. The categories are: <ul style="list-style-type: none">● Office Guidelines● Benefits● Holidays● Grievance Procedure	Category, Dialog List with the following choices: <ul style="list-style-type: none">● Office Guidelines● Benefits● Holidays● Grievance Procedure

Topic E: Create Related Forms

Lesson 10 ■ Enhancing Forms

Information to Be Gathered	Suggested Field Name and Type
The subject of the question	Subject, Text
The question	Question, Rich text
The date the question was created	QuestionDate, Date-time
The category	Category, Text

For the layer, consider setting the Height and Width properties to Auto, to accommodate varying screen sizes.

For the subform, consider including the company logo and address.

Consider adding a window title to the Question form.

2. Guidelines for the Answer form

The Answer form needs to be designed so that it is a functional form for creating an answer (response) to the questions posed using the Question form. Think about the kinds of information that would be helpful. For instance, would it be helpful to have the subject of the question on the answer form? How about a link to the question?

Also, the Answer form should have a similar look and feel as the Question form. Use the same colors and graphics used in the Question form.

The following table lists suggested information to gather, along with suggested field names and types.

Information to Be Gathered	Suggested Field Name and Type
The subject of the question	QuestionSubject, Text
The date the question was created	QuestionDate, Date-Time
A link to the question	QuestionLink, Rich text
The answer	Answer, Rich text
The date the answer was created	AnswerDate

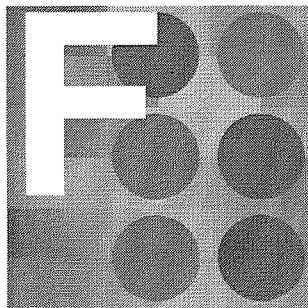
Consider adding a window title to the Answer form.

3. Create several questions and associated answers.

4. **Challenge**

Use tables to create a more pleasing, balanced look and feel for the fields and field labels.

Apply any subform you create in this activity to the Policy form as well.



Topic F: Enable AutoSave

After completing this topic, you should be able to:

- ✓ Describe the AutoSave feature.
- ✓ Describe the AutoSave database.
- ✓ Enable AutoSave.

The AutoSave Feature

AutoSave is a new feature that allows Lotus Notes documents to be saved at user-specified intervals. AutoSave can be valuable in the event that:

- The Lotus Notes client crashes.
- The user shuts down Lotus Notes improperly.
- The user loses power.

AutoSave must be enabled in two places:

- As a Form property by the database designer.
- As a User Preference by the Lotus Notes user.

When the AutoSave process is enabled, documents being created or edited are automatically copied to the AutoSave database periodically. The periodic time is set in the user's preferences. If Lotus Notes is shut down improperly, the AutoSave database contains the documents that were in process. When the user opens Lotus Notes again, the user is prompted to recover the documents.

The Autosave Database

If AutoSave is enabled, the AutoSave database is created automatically when the user creates a new document or edits an existing one. The database is local and is created from the AUTOSAVE.NSF template. The database is named as _lastname.nsf, where lastname is the first initial and last name of the user. Documents in the AutoSave database are removed when they have been properly saved, sent, or discarded. The documents cannot be opened within the AutoSave database.

Enabling AutoSave on a form

Follow these steps to enable AutoSave on a form.

Task	Procedure
1	Open the form in the Lotus Domino Designer client.
2	Open the Properties box for the form.
3	In the Options section, select Allow Autosave .
4	Save and close the form.

Enabling AutoSave in the Lotus Notes client

Follow these steps to enable AutoSave in the Lotus Notes client.

Task	Procedure
1	In the Lotus Notes client, choose File→Preferences→User Preferences .
2	In the Start Up Options section, enable AutoSave for a time frame such as every 2 minutes.
3	Click OK . Note: It is not necessary to restart Lotus Notes for the change to take effect.



Activity 10-5: Implement AutoSave



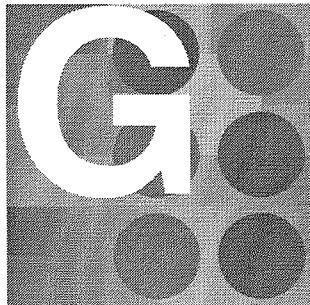
Scenario: You need to complete the following tasks to implement and test the AutoSave functionality.

1. Enable AutoSave on the form.
2. Enable AutoSave in the Lotus Notes client.
3. Test the AutoSave functionality.

Follow these steps to implement AutoSave.

Step	Action
1	Open your copy of the Practice database in Lotus Domino Designer.
2	Open the Employee Information form.
3	In the Form Properties box, on the Form Info tab, select Allow Autosave .
4	Save and close the form.
5	In the Lotus Notes client, choose File→Preferences→User Preferences .
6	In the Startup Options section, enable AutoSave for every 2 minutes .
7	Click OK .
8	Restart the Lotus Notes client.
9	In the Lotus Notes client, open the Practice database and create a new Employee Information document.
10	Fill out the form. Do not save it, but leave it open until you see the “AutoSave complete” message in the status bar. It will take about 2 minutes to appear. Leave the document open.
11	Launch the Task Manager and end the processes nlnotes.exe and ntaskldr.exe .  Note: To launch the Task Manager, press CTRL+ALT+DELETE and click Task Manager.
12	Launch the Lotus Notes client, and enter your password.
13	When prompted to recover the 1 unsaved document, click No .
14	Open the AutoSave database. You will have to enter the filename as_SNDnn.nsf in the Open Database dialog box, where nn is your student number. For example, as_SND01.nsf.
15	Right-click the document and choose Document Properties .

Step	Action
16	Switch to the Fields tab to view the fields in the document.
17	Scroll to the top of the list to see those fields specific to the AutoSave functionality.
18	Close the Properties box and the AutoSave database.
19	From the menu, choose File→AutoSave→Recover Autosaved Documents .
20	If necessary, select the document in the dialog box and click Recover .
21	Save the document properly and close it.
22	Open the AutoSave database to see that the document has been removed.
23	Verify that the AutoSave→Recover Auto Saved Documents menu option is grayed out.
24	Close all open windows.



Topic G: Implement Lotus Sametime Instant Messaging in Forms

After completing this topic, you should be able to:

- ✓ Describe how instant messaging can be integrated into Lotus Domino applications.
- ✓ List fields that can be configured to display online status.
- ✓ Enable online awareness for a field.
- ✓ Identify the characteristics of and uses for instant messaging contact lists.
- ✓ Embed an instant messaging contact list in a form.

Instant Messaging

The Lotus Instant Messaging product is a powerful tool for enabling team members who are online at the same time to communicate directly with each other. You can use Lotus Domino Designer 7 to integrate Lotus Instant Messaging into your Lotus Domino applications. Here, you will explore some of the options for including Lotus Instant Messaging in a Lotus Domino application.

The following figure shows an example of instant messaging. You can see the online status of the users shown in the From and To fields of a Lotus Notes mail document. You can right-click a name and initiate a chat with any user who is active.

	Who	Date	Time	Size	Subject
*	& Student ND06	01/09/2006	04:12 PM	762	Need to enable IM
*	& Student ND04	01/09/2006	04:12 PM	798	Need to Enable Instant Messaging on my Notes Client
*	Student ND01	01/09/2006	04:13 PM	823	Tomorrow's Meeting

Figure 10-7: Instant messaging example

For users to use instant messaging functionality, they must be connected to a Lotus Instant Messaging server through a Lotus Notes location document.



Note: Instant messaging functionality is not available for Web applications.

Online Status Display

Names, Authors, and Readers fields can all be enabled for online awareness, so that the field displays the status of users who are online at the same time. While viewing a document, users can right-click an active name to initiate a chat. The following figure shows the property to select when enabling instant messaging for a field.

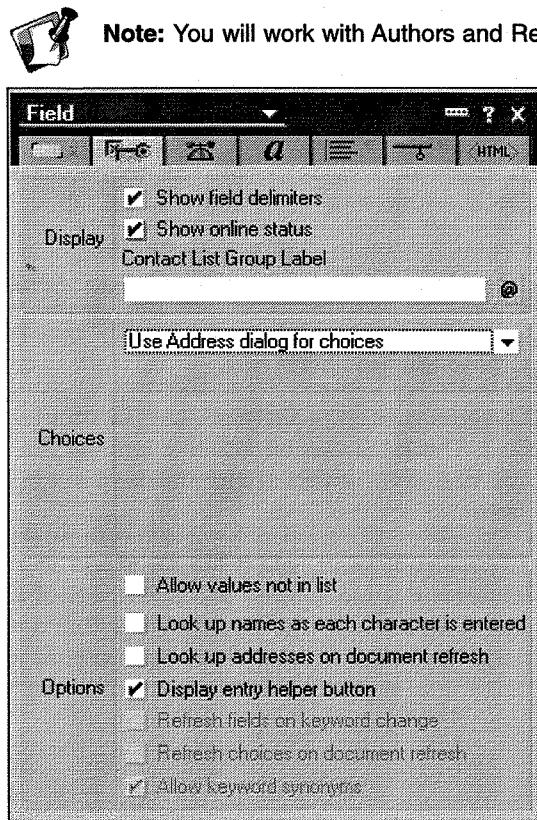


Figure 10-8: Enabling instant messaging for a field

When you open a document, if the author is online, the online indicator will appear beside the author's name, as shown in this figure.

The document form includes the following fields and information:

- Header: Worldwide CORPORATION
- Address: 123 Worldwide Blvd
Metropolis, PA 45555
- Title: Question
- Text area:
 - First Name: Student
 - Last Name: ND01
 - Contact Person:  Student ND01/WWCorp
 - Date of Question: 12/27/2005
 - Category: Office Guidelines
 - Subject: Cubicle Decorations
 - Details: How can I attach things to the walls of my cubicle without damaging them?

Figure 10-9: The online indicator

Enabling Online Awareness for a Field

To enable online awareness for a Names, Authors, or Readers field:

Task	Procedure
1	Open a form in a database in Lotus Domino Designer.
2	Create a Names, Authors, or Readers field that evaluates to an abbreviated hierarchical name, such as Doctor Notes/WWCorp.
3	Open the Control tab of the Field Properties box, and check the Show Online Status check box.
4	Optionally, enter a Contact List Group label for the users listed in the field. The label can be either a formula that evaluates to a text string, or you can enter a text string enclosed in quotation marks, such as "Sales Team".
5	To test your work, open the database in Lotus Notes. In the view that contains the form, open the document of an active author. You see the online status indicator beside the author's name. Right-click the author's name and click Chat with to open a chat window.



Optional Practice Activity 10-6: Enable Instant Messaging in a Field



Scenario: The Worldwide Corporation makes every effort to facilitate internal communication and collaboration. Sometimes, employees have questions about policies that might be too specific or sensitive to use the Question and Answer forms, or more information might be needed in order to answer the question completely.

1. On the **Question** form in your copy of the **Policies and Procedures** database, add a **Names** field that evaluates to an abbreviated hierarchical name. Name the field **Contact Person**, and have it populated by selecting the name from WWCorp's address book. Use input validation to make it a mandatory field.
2. Enable online awareness for the new field.
3. Test your changes.



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

Instant Messaging Contact List

The Instant Messaging Contact List embedded element lets a designer embed an instant messaging contact list in a page, form, subform, or rich text field of a document. This functionality allows users to see an instant messaging contact list displayed on a page or in a document and to initiate a chat.

After embedding the contact list, you can improve the display by setting size and color properties in the Embedded Contact List Properties box, as shown in this figure.

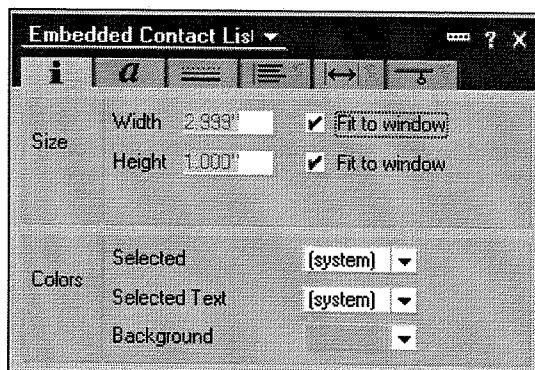


Figure 10-10: Embedded Contact List properties

Topic G: Implement Lotus Sametime Instant Messaging in Forms

Lesson 10 ■ Enhancing Forms

This figure shows how an instant messaging contact list displays in a document.

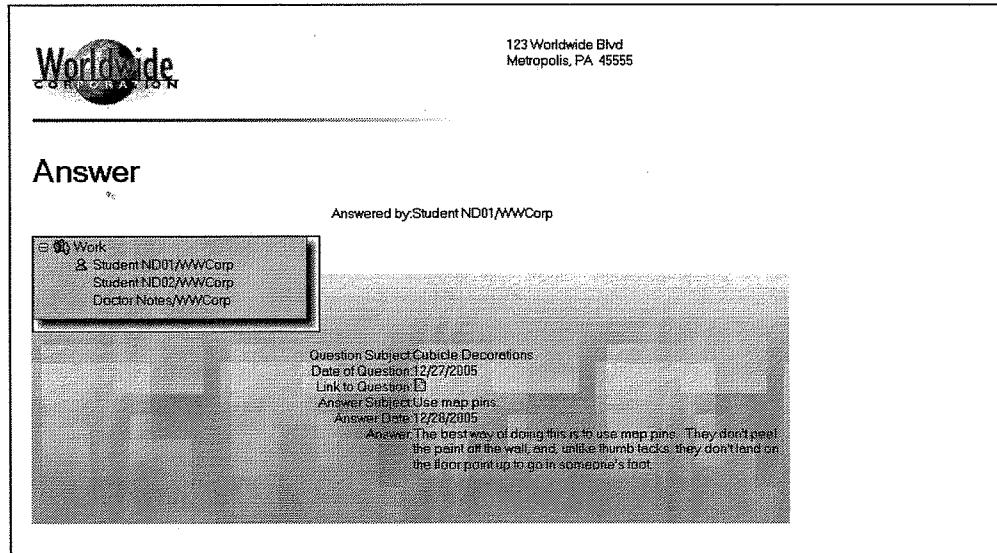


Figure 10-11: A contact list in a document

Embedding an instant messaging contact list on a form

To embed an instant messaging contact list on a form:

Task	Procedure
1	In Lotus Domino Designer, in the appropriate database, create a table on a form and set its properties for how the table displays.
2	With the cursor in the appropriate column in which you want to embed an Instant Messaging contact list, choose Create→Embedded Element→Instant Messaging Contact List .
3	Adjust the properties of the embedded contact list, as necessary to improve the display of the contact list.
4	Save the form.
5	When you test this functionality, you open a document created with this form. When you double-click the document to put it in edit mode, you should see your embedded contact list in the table.



Optional Practice Activity 10-7: Embed an Instant Messaging Contact List on a Form

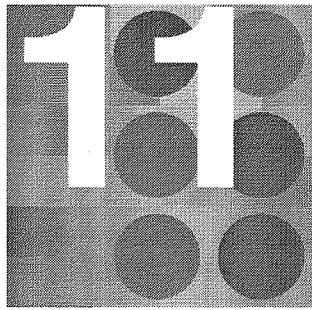


Scenario: To further facilitate communication and collaboration between Worldwide Corporation employees, you will embed an instant messaging contact list on the Answer form.

1. On the Answer form in your copy of the Policies and Procedures database, create a single-cell (1x1) fixed-width table just below the "Answered by" text in the field table. Make the cell 3 inches wide, and remove all of the table borders.
2. Embed an instant messaging contact list in the new table. Make the list fit to window for both width and height, set the background to yellow, and give it a ridge border with a drop shadow.
3. Test your changes.
4. Close all open windows.

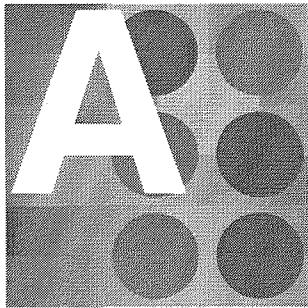


Note: The solution for this activity can be found in Appendix A, *Solutions to Practice Activities*.



Enhancing Views

- **Topic A:** Define Hierarchical Views
- **Topic B:** Display Response Data in Views
- **Topic C:** Implement Shared Columns
- **Topic D:** Enable Lotus Sametime Instant Messaging in Views
- **Topic E:** Create Views for the Web



Topic A: Define Hierarchical Views

After completing this topic, you should be able to:

- ✓ Define hierarchical views.
- ✓ Describe how to create hierarchical views.
- ✓ Describe how to display responses in a hierarchical view.

Hierarchical Views

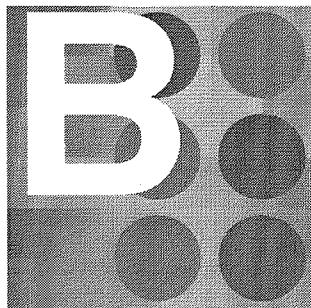
Earlier in the course, you created Question and Answer forms. Unless you design a view to display this document hierarchy, the documents are not easy to find and their relationship is not necessarily apparent.

A **hierarchical view** is a view that displays main documents and their associated response documents, and visually distinguishes between them. Each main document has its response documents indented under it. Displaying response documents beneath main documents helps to view the progression of a discussion. Each level is indented three spaces under its parent document.

Displaying documents in a hierarchy

Follow these steps to display main documents and response documents in a hierarchy.

Task	Procedure
1	Create a view.
2	Write an appropriate selection formula.
3	Set the Show response documents in a hierarchy View property.
4	Create columns to display the data.
5	Create a responses only column for displaying the response documents.
6	Save and test the view.



Topic B: Display Response Data in Views

After completing this topic, you should be able to:

- ✓ Describe what views display by default.
- ✓ Identify the characteristics of and uses for responses-only columns.
- ✓ Describe how view selection formulas can be used to display data from more than one form.

Default View Display

View columns are structured to display data from fields in the main document. The view displays little, if any, data from response documents.

If the response form happens to contain fields with the same names as those specified in the column formulas, that data will appear in the view. Otherwise, the rows representing responses show no meaningful data. To specify information to be displayed about each response document, use a responses-only column.

Responses-Only Columns

A responses-only column displays data from a document only if that document is a response or response to response document. The responses-only column should be positioned to the left of the column under which you want it to indent. The data displayed for a responses-only column starts just to the right of where you position it in the row, and extends to take up the entire row.

Note the following about placement of columns:

- Columns to the left of the responses-only column, such as a date, will display response data (if available in the document) in the row.
- Columns to the right of the responses-only column do not appear for response documents; the contents of the responses-only column appear instead.

Creating a responses-only column

Follow these steps to add a responses-only column to a view.

Task	Procedure
1	Open the view in IBM® Lotus® Domino Designer®.
2	Create a new column to the left of the column under which responses will indent.
3	On the Basics tab of the Column Properties box: <ul style="list-style-type: none">● Leave the column title blank.● Set the width to 1.● Select Show responses only.
4	Write a column formula that displays information about the response documents in the column, such as subject, authors, and creation dates.
5	Save the view.
6	Test the view by previewing it in IBM® Lotus Notes®.

Example of a formula used in a responses-only column

Typically, a responses-only column contains information from several fields. The following example formula concatenates two fields and adds static text. Notice that the Date field has been changed to text.

Subject + " - " + @Text(Date)

Multiple Forms in Views

To this point, the views you have worked with display data using only one form. When using more than one form in a view, the view selection formula must be written to select both of them. The following code is an example of a view formula that uses two forms, Main and Response:

`SELECT (Form = "Main") | (Form = "Response")`

The code could also be written as:

`SELECT Form = "Main" | @AllDescendants`



Practice Activity 11-1: Display Response Data in Views



Scenario: Now that you have created Question and Answer forms for creating Q&A documents about Worldwide's policies, you need to create a view that shows the questions with their corresponding answers in a hierarchical view.

In the following activity, you will create a hierarchical view that displays the question and answer documents.

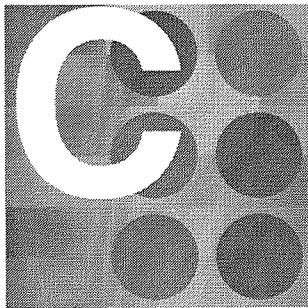
Note: Make sure you do this work in your copy of the Policies and Procedures database.



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. Create a view that displays the subject field of the question document, and the date the question was created.
2. Show the subject of the answer document, along with the date the answer was created, in a single column.
3. Display the questions by category.



Topic C: Implement Shared Columns

After completing this topic, you will be able to:

- ✓ Describe column sharing.
- ✓ Implement shared columns.
- ✓ Work with shared columns.

Shared Columns

As with fields and actions, columns can also be shared. By using a **shared column**, you eliminate the need to create the same column in multiple views or even copy and paste from one view to another. When you insert a shared column, you can choose to use the formula and properties of the column or just the formula.

Creating a new shared column

Follow these steps to create a new shared column.

Task	Procedure
1	On the Lotus Domino Designer menu, click Create → Design → Shared Column . Or, in the Design Pane, select Shared Code → Columns and click the New Shared Column button.
2	Enter a name in the Shared Column Properties box. Optionally, enter an Alias and Comments.
3	In the Properties box, switch to Column and enter a title for the shared column. Make any additional changes, as desired. All standard column properties are available for the shared column. Note: When you first create a shared column or when you open it for editing, the Shared Column properties will appear by default. You must switch to the Column properties to make changes to properties such as title and style.
4	In the Objects pane, select Column value .
5	In the Programmer's pane click Simple Function , Field , or Formula , then enter the function, field, or formula to display in the shared column. Note: When you first create a shared column or when you open it for editing, the Shared Column Properties box will appear by default. You must switch to the Column Properties to change properties such as title and style.
6	Save the column.

Creating a shared column from an existing column

Follow these steps to create a shared column based on an existing column.

Task	Procedure
1	Open the view with the desired column into the Lotus Domino Designer client.
2	Select the column to share and choose Create→Copy as shared column or right-click the column and choose Copy as shared column .
3	In the dialog box, enter a name for the new shared column.
4	Click OK to save the column. Note: The column that was used to create the shared column is not affected by any changes to the new shared column.

Using a shared column in a view

Follow these steps to insert a shared column.

Task	Procedure
1	Open the view where you want to place the shared column in the Lotus Domino Designer client.
2	Click on an existing column where you want to place the shared column.
3	From the menu, choose Create→Insert New Shared Column or Append New Shared Column .
4	Select the Use Formula Only box if you want to change the formatting of the shared column directly in this view. If selected, any formatting changes you make in this view will not affect other instances of the shared column, and any formatting changes you make to the shared column outside of this view will not be propagated to this instance of the shared column. If you do not select this box at the time of insertion, you will not be able to change the properties of the column inside this view.

Editing a shared column

Follow these steps to edit a shared column.

Task	Procedure
1	In the Design Pane, select Shared Code→Columns and double-click the name of the shared column.
2	Make any desired changes to the shared column in the Column Properties box and in the Programmer's pane.
3	Save the changes. When you save the changes, any view using the shared column will be updated. Note: If you change the name of a shared column (not the column title) in the Shared Column Properties box, any view that uses the shared column with the old name will not be updated.

Open for Editing

If a view that uses the shared column is currently open for editing in Lotus Domino Designer at the time you choose to save your changes, you will receive the following message: "A folder or view (ViewName) using this shared column is being edited. Please exit the view or folder before continuing this operation."

Note: You can also open a shared column for editing, by selecting the column in a view and choosing **Design→Edit Shared Column**; however, this process will leave the view open and when you attempt to save the changes to the shared column, you will receive the above error message.

Deleting a shared column

Follow these steps to delete a shared column.

Task	Procedure
1	In the Design pane, select Shared Code→Columns .
2	Highlight the name of the shared column to be deleted and press Delete or choose Edit→Delete . Note: When a shared column is deleted it is not removed from any views where it is used. The column reverts to a standard or unshared column.



Activity 11-2: Create shared columns

Follow these steps to create shared columns.

Step	Action
1	Open your copy of the Practice database in Lotus Domino Designer.
2	Navigate to Shared Code→Columns .
3	Click the button to create a new shared column.
4	Type SSalary as the column name
5	Switch the Properties box from Shared Column to Column .
6	Type Salary for the column title.
7	In the Programmer's pane, set the column to display the Salary field.
8	Using column properties, center the heading and the value.
9	Save and close the column
10	Open the Employee Information view.
11	Right-click the Department column, and choose Copy as Shared column .
12	Type SDepartment as the name, and click OK to save the shared column.



Activity 11-3: Insert shared columns

Follow these steps to insert the shared columns into an existing view.

Step	Action
1	In your copy of the Employee information view, delete the last two columns.
2	Select the First Name column.
3	Choose Create→Append New Shared Column .
4	Select the SDepartment column, Use Formula Only , and Insert .
5	Choose Create→Append New Shared Column .
6	Select the SSalary column and Insert .
7	Save the view.



Activity 11-4: Modify shared columns

Follow these steps to modify shared columns.

Step	Action
1	In your copy of the Employee Information view, double-click the Salary column. Notice that all of the properties are greyed out.
2	Choose Design→Edit Shared Column .
3	Switch to Column Properties and click the Advanced Format tab.
4	Change the style to Number .
5	Specify Number Format, 2 decimal places, and punctuated at thousands.
6	Right-justify the value.
7	Save and close the shared column. You will need to close the Employee Information view first.
8	Verify that the format of the Salary column in the Employee Information view uses the new format.