

## Can't create this control

You tried to add a third-party control to your dialog, but Dialog Editor couldn't find the DLL in which the control is implemented. This error is a symptom that the third-party control may not be installed on your system. Check to see that the control is installed, and that it has been installed properly. Also, check the control's pathname.

---

```
{button ,AL('H_DLG_ERROR_CANT_CREATE_STOCK_CONTROL;H_IDE_IDE_ERROR_MESSAGES_OVER;',0)}  
See related topics
```

### **Can't create stock control**

You selected one of the stock controls from the "Create... Control" menu, or pressed the Toolbox icon corresponding to a stock control, and the Dialog Editor didn't find the DLL in which the stock controls are implemented. This error is a symptom that the control or the Lotus product may not be installed on your system. Check to see that the control is installed, and that the Lotus product has been installed properly. Also, check the control's pathname.

---

{button ,AL('H\_DLG\_ERROR\_CANT\_CREATE\_CONTROL;',0)} [See related topics](#)

### **Adding a control using the Toolbox**

1. Choose View - Show Toolbox to display the Toolbox.
2. Click a control icon in the Toolbox.

**Note** To use the Toolbox to add a custom control, you must add the control icon to the Toolbox. See [Adding controls to the Toolbox](#).

3. Move the mouse pointer where you want the control to appear in the dialog box.
4. Click and drag until the control is the size you want.

**Tip** Double-click the control icon in the Toolbox to add a control to the middle of the dialog box.

---

```
{button ,AL(^;H_LDE_NAMING_A_CONTROL_STEPS;H_LDE_SETTING_CONTROL_PROPERTIES_STEPS',0)}  
See related topics
```

### **Adding a control using the Create menu**

1. Choose Create - Control.
2. Do one of the following:
  - To add a standard Lotus control, select a control from the menu.
  - To add a custom control, choose More, select a control from the list, and click OK.
3. Position and size the control in the dialog box before adding the next control.

---

{button ,AL(^;H\_LDE\_NAMING\_A\_CONTROL\_STEPS;H\_LDE\_SETTING\_CONTROL\_PROPERTIES\_STEPS',0)}  
[See related topics](#)

### **Adding custom controls to the Toolbox**

In the Toolbox, you can display the icon for any OCX custom control you have installed on your system.

1. Choose File - Toolbox Setup.
2. Select controls from the list of available controls.
3. Click OK.

**Tips** In Toolbox Setup, click Select All to select all of the controls in the list; click Deselect All to clear your selections. To display just the list of currently selected Toolbox controls, select Show selected controls only. You cannot remove the standard Lotus controls from the Toolbox.

---

{button ,AL( ;H\_LDE\_ADDING\_A\_CONTROL\_STEPS',0)} [See related topics](#)

**Details: Aligning multiple controls**

The table below summarizes the alignment options.

<u>Alignment option</u>	<u>Does this</u>
Left	Aligns all controls with the left edge of the active control.
Right	Aligns all controls with the right edge of the active control.
Top	Aligns all controls with the top edge of the active control.
Bottom	Aligns all controls with the bottom edge of the active control.
Center Horizontally	Aligns the center-top handles of all controls with the center-top handles of the active control.
Center Vertically	Aligns the center-side handles of all controls with the center-side handles of the active control.

---

{button ,AL(`H\_LDE\_ALIGNING\_CONTROLS\_STEPS',1)} [Go to procedure](#)

## Aligning multiple controls

You can align multiple controls according to the placement of the control that is active.

1. Select more than one control.

Make the control you want to align the others to the active control. If it is active, it will have black handles.

2. Choose Dialog - Align.
3. Choose the alignment you want.

**Tip** You can restore the controls to their original positions by choosing Edit - Undo.

---

{button ,AL(`H\_LDE\_ALIGNING\_CONTROLS\_DETAILS',1)} [See details](#)

{button ,AL(`;H\_LDE\_SELECTING\_AND\_DESELECTING\_CONTROLS\_OVER',0)} [See related topics](#)

## Overview: Connecting Help

When you connect Help to a dialog box, you can press F1 and display a Help topic about the dialog box and about each active control.

You do the following to connect Help topics to dialog boxes and controls:

- Write the Help topics.
- Map each topic to a context number and create a compiled Help file, using a Help compiler.
- Set the Help file name property of the dialog box to the name of the compiled Help file, using the InfoBox in the Dialog Editor.

**Tip** To display Help topics in a secondary Help window, append the secondary window name to the Help file name. For example, "Myhelp.hlp>Steps" displays topics in the "Steps" secondary Help window.

- Assign a Help context ID to the dialog box, using the InfoBox in the Dialog Editor.
- (Optional) Assign a Help context ID to each individual control in the dialog box, using the InfoBox in the Dialog Editor.

The context ID you assign to the dialog box serves as the default Help topic. If the active control has a Help context ID assigned to it, pressing F1 displays that topic. Otherwise, pressing F1 displays the default Help topic that is assigned to the dialog box.

---

{button ,AL(^;H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)



**Details: Copying a control**

When you make a copy of a control within the same dialog box, the caption of the control remains the same. However, the Dialog Editor assigns a new name, such as "Text2," to each copied control.

You can change the name of a control by selecting it and using the Basics tab in the InfoBox .

**Tip** Whenever possible, it's helpful to use the parameter "Source" to refer to controls in event scripts, rather than the specific control name. When you use "Source," you do not have to change the names of the controls in the scripts you copy. If you use the specific control name in a script and then change the name, you need to find and replace every occurrence of the old name.

---

{button ,AL('H\_LDE\_COPYING\_CONTROLS\_STEPS',1)} [Go to procedure](#)

## Copying a control

Copying and pasting a control lets you easily add a group of similar controls to a dialog box. If you have written scripts for the controls, you can copy those scripts with the control.

1. Select the controls you want to copy.
2. Choose Edit - Copy.



3. Choose a command:
  - To paste the controls, including properties but not scripts, choose Edit - Paste.
  - To paste the controls, including properties and scripts, choose Edit - Paste Special, and click Yes in the Paste Special dialog box.

**Note** When you paste controls to the same dialog box, the Dialog Editor pastes them directly on top of the copied controls.

4. Drag the copies to where you want to place them on the dialog box.

---

{button ,AL(`H\_LDE\_COPYING\_CONTROLS\_DETAILS',1)} [See details](#)

{button ,AL(`;H\_LDE\_SELECTING\_AND\_DESELECTING\_CONTROLS\_OVER',0)} [See related topics](#)

## Creating a dialog box

When you create a dialog box, the Dialog Editor adds a tab in the Dialog Editor window. The tab contains a new dialog box with default OK and Cancel buttons.

- Choose Create - Dialog.



**Note** The Dialog Editor always displays at least one tab with one dialog box. If you delete this dialog box, the Dialog Editor automatically creates a new dialog box to replace it. You cannot delete a dialog box using Undo.

---

{button ,AL( ;H\_LDE\_DELETING\_A\_DIALOG\_BOX\_STEPS;H\_LDE\_NAMING\_A\_DIALOG\_BOX\_STEPS;H\_LDE\_SAVING\_DIALOG\_BOXES\_STEPS',0)} [See related topics](#)

**Deleting a dialog box**

1. Click the tab for the dialog box you want to delete.
2. Choose Edit - Delete Dialog.

Check your scripts for any code that refers to the deleted dialog box and remove or modify the code appropriately.

### **Deleting a control**

1. Select one or more control(s).
2. Choose Edit - Clear.



**Tip** You can also select the control(s) and press DEL.

## Overview: Design time and run time

You can set and change a dialog box and its control properties at design time or run time.

### Design time

Use the InfoBox in the Dialog Editor to set design-time properties of a dialog box and its controls. Design-time properties determine how the dialog box appears when you first display it, unless you change the properties at run time.

### Run time

To change the run-time properties of a dialog box or control, add script to a dialog box or control event or call a subroutine that changes the dialog box or control properties.

For example, you can add script to the dialog box Load event to set initial values for variables connected to dialog box controls. The following script changes the caption of the MyDialog dialog box to "Employee Names," initializes the Rate variable to 500, and displays this value as a string in the txtAmount text box.

```
'Setting dialog box properties at load time
Sub Load(Source As Lotusdialog)
    Source.Caption = "Employee Names"
    Rate = 500
    Source.txtAmount.text = Str(Rate)
End Sub
```

### Read-only properties

Some properties are read-only, and cannot be directly changed at either design or run time. For example, the ListCount property of the LotusComboBox control is read-only. It always equals the number of items in the list.

---

{button ,AL(`;H\_LDE\_DESIGN\_TIME\_LOAD\_TIME\_AND\_RUN\_TIME\_OVER;H\_LDE\_MODAL\_AND\_MODELESS\_DIALOGS\_OVER;H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER;H\_LDE\_SCOPE\_AND\_VISIBILITY\_OF\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)

**Dialog and control names**

The name of a dialog or control must be a valid LotusScript identifier.

The following rules govern the construction of identifiers in a script:

- The first character in an identifier must be an uppercase or lowercase letter.
- The remaining characters must be letters, digits, or underscore ( \_ ).
- The maximum length of an identifier is 40 characters, not including the optional suffix character.
- Names are case-insensitive. For example, VerED is the same name as vered.
- Characters with ANSI codes higher than 127 (that is, those outside the ASCII range) are legal in identifiers.

**Note** Do not include a data type suffix character (% , & , ! , # , @ , or \$ ) in a control name.

## Overview: Displaying dialog boxes in a script application

Use the Show method to display a dialog box in a script application. The Style argument of the Show method determines whether the dialog box is invoked as modal, modeless, or dockable.

**Note** In SmartSuite Release 9, Word Pro is the only application which supports dockable dialogs.

By default, the Show method invokes dialog boxes as modeless. For example, the following statement invokes MyDialog as a modeless dialog box:

```
MyDialog.Show
```

To invoke a dialog box as modal, set the Style argument for the Show method to 1, or use the constant DIALOG\_MODAL, as shown in the example below.

```
MyDialog.Show DIALOG_MODAL
```

To invoke a dialog box as dockable, set the Style argument for the Show method to 2, or use the constant DIALOG\_DOCKABLE, as shown in the example below.

```
MyDialog.Show DIALOG_DOCKABLE
```

---

{button ,AL('H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER;H\_LDE\_USING\_CONSTANTS\_OVER',0)} [See related topics](#)



## **Closing the Dialog Editor**

When you close a Dialog Editor session, the document that contains the dialog boxes remains active. You must save the document to save any changes you made to the dialog boxes.

- Choose File - Close Dialog Editor from the Dialog Editor menu.

---

{button ,AL(`H\_LDE\_STARTING\_DIALOG\_EDITOR\_STEPS;H\_LDE\_SAVING\_DIALOG\_BOXES\_STEPS',0)} See  
related topics

## Details: Entering and editing scripts

### Using the Dialog Editor with the LotusScript IDE

The Dialog Editor is part of the LotusScript Integrated Development Environment (IDE).

When you open the IDE Script Editor for a dialog box, the Script Editor displays one of the following:

- If the dialog box or the selected control contains a script, the Script Editor shows either the last script you edited or the first sub, function, or other group of statements that contains a script for the control or dialog box.
- If the dialog box or selected control does not contain a script that you have edited, the Script Editor shows the sub for the object's default event.

### Locating Lotus dialog box script classes in the IDE Browser panel

The IDE Browser panel lists standard Lotus dialog box script classes under the class list for each Lotus product. For example, if you are creating a dialog box in 1-2-3, select "Lotus 1-2-3: Classes" in the Categories drop-down list and scroll down to display the available Lotus dialog box classes.

### Entering scripts for custom controls

You enter scripts for custom OCX controls in the same way you enter scripts for standard Lotus controls. After you add a custom control to a dialog box, the IDE Object drop-down lists the name of the custom control under the class list for the Lotus product you are using. For example, if you are creating a dialog box in Word Pro and add a custom control, the IDE adds information about the custom control under "Lotus Word Pro: Classes."

**Note** The Browser uses the name for a custom control that appears in the title bar of the InfoBox. You may need to refresh the class list in the Browser by switching to a different category, then switching back, to see the custom control in the class list.

All of the properties, methods, and events available in the LotusControl class are automatically added to any custom control you add to a dialog box. Any limits of the LotusControl class properties take precedence over any overlapping property of the custom control. For example, if the custom control allows 256-character captions but LotusControl limits captions to 40 characters, the 40-character limit takes precedence.

You can script all of the custom control class members, as well as all of the LotusControl base class members. However, some base class members may not apply to each custom control. For example, if a custom spinner control does not display a caption, using the InfoBox or script to set the LotusControl Caption property for the spinner has no visible effect.

### Saving dialog box scripts

The scripts you create in the IDE are saved in the product document, along with the dialog box. For example, if you create a dialog box in Word Pro and use the IDE to add scripts to dialog box controls, the dialog box and the scripts are saved in the current .LWP file.

### Default events for standard Lotus dialog box controls

The table below lists the default event for each standard Lotus dialog box control.

<u>Lotus Control</u>	<u>Default event</u>
Dialog	Load
Text box	Change
Combo box	Pick
Label, check box, option button, list box, command button, slider, image	Click
Frame, progress bar	Gotfocus
Spin button	Spindown

---

{button ,AL(^H\_LDE\_ENTERING\_AND\_EDITING\_SCRIPTS\_STEPS',1)} [Go to procedure](#)

## Entering and editing scripts

You use the IDE Script Editor to write and edit scripts, check script syntax, and set breakpoints for debugging dialog box scripts.

- In the Dialog Editor, double-click the control you want to script, or double-click any open area to script the dialog box itself.

The Script Editor displays a script for the dialog box or selected control in the Script Editor.

---

{button ,AL(`H\_LDE\_ENTERING\_AND\_EDITING\_SCRIPTS\_DETAILS',1)} [See details](#)

{button ,AL(`;H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)

## Overview: Collecting user input

You often use dialog boxes to prompt users for information. You use scripts attached to the dialog box and its controls to:

- Gather the information entered in a control
- Change the status of other controls in the dialog box
- Validate the information entered
- Pass information to other procedures in the script application

For example, the procedure below is attached to the Click event of the OK button in MyDialog, a dialog box in a 1-2-3 script application. The procedure enters the text in the txtEmployeeName text box into the EmployeeNameCell range in 1-2-3.

```
'Getting user input
Sub Click(Source As Lotuscommandbutton)
    Dim EmployeeName As String
    EmployeeName = MyDialog.txtEmployeeName.text
    [EmployeeNameCell].Select
    Selection.Contents = EmployeeName
End Sub
```

## Validating user entries

You can also attach scripts to the change events of controls to validate user entries.

For example, the procedure below checks that the entry in the txtRate text box is between 0 and 100.

```
'Procedure to validate user entries
Sub Change(Source As Lotustextbox)
    Rate = Val(Source.Text)
    If Rate < 0 Or Rate > 100 Then
        MsgBox "Bad Rate"
        Source.Text = "0"
    End If
End Sub
```

---

{button ,AL(`;H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)

**To move, collapse, or close the InfoBox**

- To move the InfoBox, drag here
- To collapse or reopen the InfoBox, double-click here
- To close the InfoBox, click here



## Overview: Dialog box styles

A script application can invoke a dialog box as modal, modeless, or, in some applications, dockable. The Style argument of the Show method determines how the dialog box is invoked.

**Note** In SmartSuite Release 9, Word Pro is the only application which supports dockable dialogs.

### Modal dialog boxes

A modal dialog box requires the user to complete or cancel an action before continuing. For example, when you open a file, you use a modal dialog box where you must specify and open a file, or cancel the dialog box, before you can do anything else.

Some key points about scripting a modal dialog box include the following:

- Modal dialog boxes are always enabled and visible. Deselecting the Enabled or Visible properties in the InfoBox has no effect when you invoke a dialog box modally. Attempting to turn off the Enabled or Visible properties of a modal dialog box in a script results in a run-time error.
- A script in a modal dialog box can invoke another modal dialog box, but not a modeless dialog box.
- A script that closes a modal dialog box also closes any other modal dialog boxes the original modal dialog box invoked.
- Modal dialog boxes usually include OK and Cancel buttons, with scripts that process the information entered in the dialog box and close it.

### Modeless dialog boxes

A modeless dialog box lets the user change settings in the dialog box and perform other tasks in the application without closing the dialog box. For example, the InfoBox in Lotus products is modeless. You can select and make many changes to several objects without closing the InfoBox.

Some key points about scripting a modeless dialog box include the following:

- You can hide a modeless dialog box by setting the Visible property to False or using the Hide method.
- You can disable a modeless dialog box by turning off the Enabled property.
- A script in a modeless dialog box may invoke another dialog box as either modeless or modal.
- Closing a modeless dialog box has no effect on other dialog boxes invoked from it.
- Modeless dialog boxes may include an Apply button, with scripts that process the information entered in the dialog box and keep the dialog box displayed and ready to accept more information.

### Dockable dialog boxes

A dockable dialog box is similar to a modeless dialog box in so far as it lets the user change settings in the dialog box and perform other tasks in the application without closing the dialog box. The key difference is the fact that a dockable dialog box is affixed to one side of a window and cannot be moved from that side. For example, the Spell Checking bar in Lotus Word Pro is a dockable dialog box. You can select and make many changes to several objects without closing the Spell Checker but you cannot move the Spell Checker around on the screen as you could a modal or modeless dialog box.

**Note** In SmartSuite Release 9, Word Pro is the only application which supports dockable dialogs.

Some key points about scripting a dockable dialog box include the following:

- The DockTo and DockType properties are read only and thus cannot be set from a script.
- You can hide a dockable dialog box by setting the Visible property to False or using the Hide method.
- You can disable a dockable dialog box by turning off the Enabled property.
- A script in a dockable dialog box may invoke another dialog box in the modal, modeless, or dockable style.
- Closing a dockable dialog box has no effect on other dialog boxes invoked from it.
- Dockable dialog boxes may include an Apply button, with scripts that process the information entered in the dialog box and keep the dialog box displayed and ready to accept more information.

---

```
{button ,AL('H_LDE_DESIGN_TIME_LOAD_TIME_AND_RUN_TIME_OVER;H_LDE_PROGRAMMING_DIALOG_BOXES_OVER;H_LDE_SCOPE_AND_VISIBILITY_OF_DIALOG_BOXES_OVER;H_LDE_DISPLAYING_DIALOG_BOXES_IN_YOUR_APPLICATION_OVER',0)} See related topics
```



## Details: Moving controls

### Using the grid

You can automatically align controls to a dialog box grid. For more information, see [Using the grid](#).

### Using the keyboard

Select a control, press and hold down CTRL, then press UP, DOWN, RIGHT, or LEFT to move a control one pixel or grid unit in any direction.

### Using the InfoBox

You can use the InfoBox to position a control.



Select the control and click the size and position tab. Enter new values for the top left corner of the control.

---

{button ,AL(`H\_LDE\_MOVING\_CONTROLS\_STEPS',1)} [Go to procedure](#)

{button ,AL(`;H\_LDE\_USING\_THE\_INFOBOX\_STEPS',0)} [See related topics](#)



### **Moving controls front and back**

When a control overlaps another control, you can display it either in front or behind the overlapping control.

1. Select a control that overlaps another control.
2. Choose a command:
  - To move the selected control in front of the overlapping control, choose Dialog - Bring to Front.
  - To move the selected control behind the overlapping control, choose Dialog - Send to Back.

## Moving controls

You can drag controls to move them.

1. Select one or more controls.
2. Drag the control(s) by any part except the handles.
3. Release the mouse button.

**Tip** If you select multiple controls, you can move them all by dragging one.

---

{button ,AL(`H\_LDE\_MOVING\_CONTROLS\_DETAILS',1)} [See details](#)

{button ,AL(`H\_LDE\_USING\_THE\_GRID\_STEPS;H\_LDE\_SELECTING\_AND\_DESELECTING\_CONTROLS\_OVER',0)} [See related topics](#)

### **Moving a control to another dialog box**

You can cut controls from one dialog box and move them to another dialog box in the same document by cutting and pasting.

1. Select the controls want to move.
2. Choose Edit - Cut.



3. Click the tab of the dialog box where you want to place the controls.
4. Choose a command:
  - To paste the controls, including properties but not scripts, choose Edit - Paste.
  - To paste the controls, including properties and scripts, choose Edit - Paste Special, and click Yes in the Paste Special dialog box.
5. Drag the copies to where you want to place them on the dialog box.

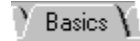
---

{button ,AL(`;H\_LDE\_COPYING\_CONTROLS\_STEPS',0)} [See related topics](#)

## Naming a control

Naming a control makes it easier to code scripts. The Dialog Editor assigns a default name, such as "Text1," to every control, but you can use the InfoBox to change the name.

1. Right-click the control.
2. Choose Properties.
3. Click the Basics tab in the InfoBox.



4. Enter a name for the control.
5. (Optional) Move, collapse, or close the InfoBox.

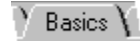
---

```
{button ,AL(^;H_LDE_ADDING_A_CONTROL_STEPS;H_LDE_ADDING_A_CONTROL_USING_THE_CREATE_MENU_STEPS;H_LDE_ENTERING_AND_EDITING_SCRIPTS_DETAILS;H_LDE_NAMING_A_DIALOG_BOX_STEPS;H_LDE_SCOPE_AND_VISIBILITY_OF_DIALOG_BOXES_OVER;H_LDE_USING_THE_INFOBOX_STEPS',0)} See related topics
```

## Naming a dialog box

Naming a dialog box makes it easier to code scripts. The Dialog Editor assigns a default name and caption, such as "Dialog1," to every dialog box. You can use the InfoBox to change the name.

1. Right-click any open area in the dialog box.
2. Choose Properties.
3. Click the Basics tab.



4. Enter a name for the dialog box.
5. Enter a caption.

**Tip** You use the dialog box name to refer to the dialog box in script procedures. The caption can be a string variable of up to 256 characters in length. It appears in the title bar when you display the dialog box in your application.

---

```
{button ,AL(^;H_LDE_CREATING_A_DIALOG_BOX_STEPS;H_LDE_ENTERING_AND_EDITING_SCRIPTS_STEP  
S;H_LDE_NAMING_A_CONTROL_STEPS;H_LDE_SCOPE_AND_VISIBILITY_OF_DIALOG_BOXES_OVER',0)}
```

[See related topics](#)

## Details: Navigating between dialog boxes

### Related icons

You can use the following icons to navigate between dialog boxes:



Displays the Go To Dialog dialog box.



Scrolls dialog box tabs to bring hidden tabs into view.

### Related commands and keyboard shortcuts

You can use the following menu commands and keyboard shortcuts to navigate between dialog boxes:

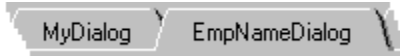
<u>Command</u>	<u>Keyboard shortcut</u>	<u>Does this</u>
View - Next Dialog	CTRL+PGDN	Goes to the dialog box in the tab to the right of the current tab.
View - Previous Dialog	CTRL+PGUP	Goes to the dialog box in the tab to the left of the current tab.
View - Go To Dialog	None	Displays the Go To Dialog dialog box.

---

{button ,AL(`H\_LDE\_NAVIGATING\_BETWEEN\_DIALOGS\_STEPS',1)} [Go to procedure](#)

## Navigating between dialog boxes

To navigate to a dialog box, click the dialog box tab.



If the dialog box you want to navigate to isn't visible, use View - Go To Dialog to find it or click a left or right scroll button.

---

{button ,AL('H\_LDE\_NAVIGATING\_BETWEEN\_DIALOGS\_DETAILS',1)} [See details](#)

## Overview: The LotusScript Dialog Editor

The LotusScript Dialog Editor provides a usable and powerful set of tools for creating and scripting dialog boxes for use in Lotus products. If you have worked with IBM VisualAge or Microsoft Visual Basic, you will be comfortable creating dialog boxes with the LotusScript Dialog Editor.

The LotusScript Dialog Editor is an OCX container, so you can take advantage of any OCX custom controls you are already using.

## Using the Dialog Editor with Lotus products

The following Lotus products support the Dialog Editor:

- 1-2-3
- Freelance Graphics
- Approach
- Word Pro

Several of the ways you use the Dialog Editor with Lotus products make it different from other development environments you may have used:

- To run the Dialog Editor, you must first run a Lotus product and then open a document in that product.
- For each document that is active in your product, you can display a Dialog Editor window that, in turn, displays the dialog boxes in that document. For example, if you are running 1-2-3 and have opened three 1-2-3 workbooks, you can display a Dialog Editor window for each of those documents.
- The Dialog Editor saves your dialog boxes in product documents. If you develop dialog boxes for a Word Pro document named INVEST.LWP, the Dialog Editor stores all the dialog boxes in INVEST.LWP when you save that document.

## Working with the Dialog Editor and LotusScript IDE together

The Dialog Editor is closely connected with the tools in the LotusScript Integrated Development Environment (IDE). Use the Dialog Editor to construct the visual layout of your dialog boxes; use the IDE to write and debug scripts that use the dialog boxes you create in the Dialog Editor.

Use the Dialog Editor for these tasks:

- Creating and naming new dialog boxes
- Adding and arranging controls in a dialog box
- Setting design-time properties for dialog boxes and controls, such as name, caption, and font
- Saving dialog boxes in product documents

Use the LotusScript IDE for these tasks:

- Entering and editing scripts for dialog boxes and controls
- Creating scripts to collect and manage user input
- Connecting help to dialog boxes and controls
- Testing and debugging dialog box scripts

---

```
{button ,AL( ;H_LDE_ENDING_DIALOG_EDITOR_STEPS;H_LDE_STARTING_DIALOG_EDITOR_STEPS;H_IDE_T  
HE_LOTUSSCRIPT_IDE_OVER;',0)} See related topics
```



## Overview: Programming dialog boxes

Once you have created a dialog box with controls in the Dialog Editor, you can use the IDE to attach LotusScript procedures to those controls and write procedures that include the dialog box in a script application.

You use LotusScript procedures for the following tasks:

- Displaying a dialog box in a script application.
- Setting the display behavior of the dialog box as modal, modeless, or dockable.
- Defining variables and integrating the controls with the dialog box.  
For example, you can define variables that hold the default entries in text boxes or the items in a list box.
- Collecting and validating user input.  
For example, you can display a dialog box with a field for phone numbers, and then use script procedures to check that the user makes valid numeric entries.
- Responding to user actions.  
For example, you can include scripts that calculate a value when a user clicks a command button or fills in a text box when a user positions a slider control.
- Integrating the dialog box and its controls with the host product.  
For example, you can include scripts to make entries in a 1-2-3 range, based on values entered in a dialog box.
- Integrating the dialog box and its controls with other dialog boxes in your script application.  
For example, a data entry application may have a series of dialog boxes that prompt users for different types of data.
- Providing Help about the dialog box and its controls.

---

{button ,AL(';H\_LDE\_DESIGN\_TIME\_LOAD\_TIME\_AND\_RUN\_TIME\_OVER;H\_LDE\_MODAL\_AND\_MODELESS\_DIALOGS\_OVER;H\_LDE\_SCOPE\_AND\_VISIBILITY\_OF\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)

## Running a dialog box from the Dialog Editor

You can run a dialog box from within the Dialog Editor to test its appearance, layout, and basic functionality.

1. Click the tab for the dialog box you want to run.
2. Choose Dialog - Run Dialog.



3. To close the dialog box that's running, do one of the following:

- Click the Close box in the dialog box title bar



- Press ALT+F4
- Press CTRL+BREAK

**Note** With this method, the Dialog Editor runs dialog boxes modally. To run a modeless or dockable dialog box, you must run a script that uses the Show method to display the dialog box. In SmartSuite Release 9, Word Pro is the only application which supports dockable dialogs.

---

{button ,AL(;H\_LDE\_DISPLAYING\_DIALOG\_BOXES\_IN\_YOUR\_APPLICATION\_OVER',0)} [See related topics](#)

### **Saving a dialog box**

You save a dialog box in the product document where you created it. For example, if you create a dialog box while working in 1-2-3, you save changes to a workbook (.123) document.

- Choose File - Save Dialogs from the Dialog Editor menu.



**Tip** Anytime you save a product document, you save all dialog boxes and scripts stored in it. You can also save any dialog boxes associated with the active document by choosing File - Save from the IDE menu.

---

{button ,AL( ;H\_LDE\_ENTERING\_AND\_EDITING\_SCRIPTS\_STEPS',0)} [See related topics](#)

## Overview: Scope and visibility of dialog boxes and controls

The Dialog Editor registers each instance of a dialog box as a named object of the LotusDialog class. It registers dialog box controls as named objects of the LotusControl class. Each dialog box also contains a LotusControls collection class that represents all of its controls. This makes the dialog box and its controls visible to all scripts contained in the document that contains the dialog box.

The Dialog Editor registers each control you add to a dialog box as an extended property. This allows you to access the control in a script by the control name.

## Referring to dialog boxes and controls in script

Use the dialog box name to refer to a dialog box in a script. To refer to a control, use the dialog box name and the control name.

For example, the following script displays "Allison" in the text box named txtName when you click a command button in the dialog box.

```
'Referring to dialog boxes and controls in script
Sub Click(Source As Lotuscommandbutton)
    MyDialog.txtName.Text = "Allison"
End Sub
```

A script in one dialog box can refer to controls in any other dialog box in the same document.

---

{button ,AL(;H\_LDE\_DESIGN\_TIME\_LOAD\_TIME\_AND\_RUN\_TIME\_OVER;H\_LDE\_MODAL\_AND\_MODELESS\_DIALOGS\_OVER;H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER;H\_LDE\_SCOPE\_AND\_VISIBILITY\_OF\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)

## Selecting and deselecting controls

The table below describes how to select and deselect dialog box controls in the Dialog Editor.

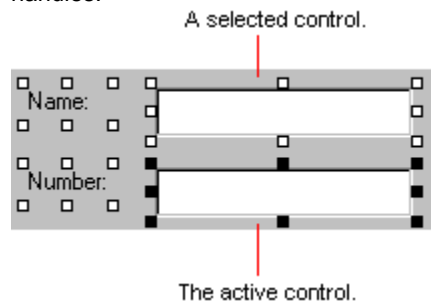
<u>Task</u>	<u>Do this</u>
Select a single control	Click the control. <b>Note</b> If you select a collection of controls, you must deselect all controls before you can select a single control.
Select a collection of controls	Move the mouse pointer near the first control, click, then drag the box to enclose the controls you want to select.
Add a single control to a collection	SHIFT+click the control.
Make a control the active control in a collection	Click the control.
Select all controls	Choose Edit - Select All.
Deselect all controls	Choose Edit - Deselect All.

## The appearance of selected controls

When you select a single control, handles appear around it.



When you select a collection of controls, the active control has black handles and the other controls have white handles.



## Keyboard shortcuts

You can deselect all controls and select an adjacent single control by pressing UP, DOWN, RIGHT, LEFT, TAB, or SHIFT+TAB.

## Related SmartIcons



Select all controls



Deselect all controls

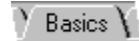
---

{button ,AL(`H\_LDE\_ALIGNING\_CONTROLS\_STEPS;H\_LDE\_SPACING\_CONTROLS\_STEPS;H\_LDE\_MOVING\_C  
ONTROLS\_STEPS',0)} [See related topics](#)

## Details: Setting control properties

### InfoBox tabs for control properties

Each of the following tabs in the InfoBox contains a category of properties you can change for each control:



Change basic properties.



Set the size and position of the dialog box on the screen.

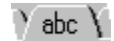


Change font, size, color, and attributes for selected text.

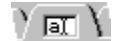


Change color, pattern, and line style.

For standard Lotus controls, the InfoBox also displays one of the following tabs, depending on the selected control:



Change properties of labels.



Change properties of text boxes.



Change properties of check boxes.



Change properties of frames.



Change properties of option buttons.



Change properties of list boxes.



Change properties of combo boxes.



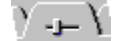
Change properties of command buttons.



Change properties of image objects.



Change properties of spin buttons.



Change properties of sliders.



Change properties of progress bars.

### Setting properties of custom controls

Custom controls use the standard control properties in the InfoBox plus additional custom properties. To set custom properties for custom controls, display the Basics tab in the InfoBox and click Properties.

---

{button ,AL(`H\_LDE\_SETTING\_CONTROL\_PROPERTIES\_STEPS',1)} [Go to procedure](#)

## Setting control properties

Use the InfoBox to set design-time properties of a control. You can make changes to one control or several controls without closing the InfoBox.

1. Right-click the control.
2. Choose Properties.
3. Click the tab for the properties you want to change.
4. Select one or more options.
5. (Optional) Move, collapse, or close the InfoBox.

**Tip** You can also choose Dialog - <control-type> Properties, press ALT+ENTER, or use the InfoBox icon to display the InfoBox.



---

{button ,AL(`H\_LDE\_SETTING\_CONTROL\_PROPERTIES\_DETAILS',1)} [See details](#)

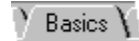
{button ,AL(`H\_LDE\_USING\_THE\_INFOBOX\_STEPS;H\_LDE\_DESIGN\_TIME\_LOAD\_TIME\_AND\_RUN\_TIME\_OVER',0)} [See related topics](#)



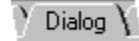
## Details: Setting dialog box properties

### InfoBox tabs

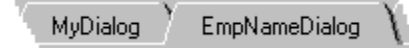
Each tab in the InfoBox contains a category of properties you can change for the current dialog box.



Changes basic properties



Sets Always on top, redraw, Control box, border style, dockable options, and help file name



Sets the size and position of the dialog box on the screen

### Size

You can set the width and height of a dialog box in either of the following ways:

- Selecting the dialog box and dragging a handle in the direction you want to size it
- Setting the Width and Height in the InfoBox

### Position

Enter values for Left and Top in the InfoBox for where the dialog box will appear when you run it. The values you enter are relative to the top left of the screen.

### Control Box

By default, dialog boxes include Control-menu and Close buttons in the title bar. To remove these buttons from a dialog box design, deselect "Control box" in the Dialog tab of the InfoBox.

If you create a dialog box without a control box, you need to provide a way to close the dialog box. For example, you might include the following code with the click events for the OK and Cancel buttons:

```
Source.Parent.Close
```

**Note** You can always close a modal dialog box by pressing CTRL+BREAK.

### When Shown Docked

You can set two options for dockable dialog boxes: DockType and DockTo. These options are used only when the dialog box is shown as dockable. DockType specifies whether the dialog box will be docked at the top, bottom, left, or right side of the window. The DockTo option specifies whether the dialog box will be docked to the application window or the document window. Neither option has any affect when the dialog box is shown as modeless or modal.

**Note** In SmartSuite Release 9, Word Pro is the only application which supports dockable dialogs.

### Border Style

Set the border style to one of the following:

<b>Border Style</b>	<b>Description</b>
No Border	Creates a borderless dialog box with no caption
Fixed Single Border	Creates a dialog box with an unsizeable border
Sizeable Border	Creates a dialog box with a sizeable border
Fixed Dialog Border	Creates a dialog box with an unsizeable border
Fixed Tool Border	Creates a dialog box with an unsizeable border and a small title bar with no Control-menu button
Sizeable Tool Border	Creates a dialog box with a sizeable border and a small title bar with no Control-menu button

Dialog boxes shown as dockable always display with no border, regardless of the border style you choose or set prior

to showing the dialog box.

---

{button ,AL('H\_LDE\_SETTING\_DIALOG\_BOX\_PROPERTIES\_STEPS',1)} [Go to procedure](#)

{button ,AL('H\_LDE\_CONNECTING\_HELP\_OVER;H\_LDE\_NAMING\_A\_DIALOG\_BOX\_STEPS;H\_LDE\_DESIGN\_T  
IME\_LOAD\_TIME\_AND\_RUN\_TIME\_OVER',0)} [See related topics](#)

## Setting dialog box properties

Use the InfoBox to set design-time properties of a dialog box.

1. Click the tab of the desired dialog box.
2. Right-click any open area in the dialog box.
3. Choose Properties.
4. Click the tab for the properties you want to change.
5. Select one or more options.
6. (Optional) Move, collapse, or close the InfoBox.

**Tip** You can also choose Dialog - Dialog Properties, press ALT+ENTER, or use the InfoBox icon to display the InfoBox.



---

{button ,AL(`H\_LDE\_SETTING\_DIALOG\_BOX\_PROPERTIES\_DETAILS',1)} [See details](#)

{button ,AL(`;H\_LDE\_SETTING\_CONTROL\_PROPERTIES\_STEPS;H\_LDE\_USING\_THE\_INFOBOX\_STEPS',0)}  
[See related topics](#)

## Sizing multiple controls

You can size multiple controls equally, according to the height or width of the control that is active.

1. Select multiple controls.
2. Choose Dialog - Equal Sizing.
3. Choose an option:
  - "Vertical" changes the height of all of the controls to the height of the active control.
  - "Horizontal" changes the width of all of the controls to the width of the active control.

**Tip** After resizing multiple controls, you can restore the controls to their original sizes by choosing Edit - Undo.

---

```
{button ,AL('H_LDE_SIZING_SINGLE_CONTROLS_STEPS;H_LDE_SELECTING_AND_DESELECTING_CONTROL_OVER;H_LDE_USING_THE_GRID_STEPS',0)} See related topics
```

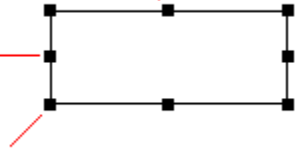
### Details: Sizing a single control

You can drag different handles to size a control.

Drag a top or bottom mid-point handle to size height

Drag a side mid-point handle to size width

Drag a corner handle to size both height and width

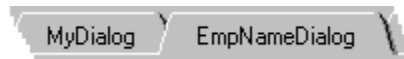


### Restoring the size of a control

After releasing the mouse button, you can restore a control to its original size by choosing Edit - Undo.

### Using the InfoBox to size a control

You can use the InfoBox to size a control.



Select the control and click the size and position tab. Enter new values for the width and height of the control.

---

{button ,AL('H\_LDE\_SIZING\_SINGLE\_CONTROLS\_STEPS',1)} [Go to procedure](#)

{button ,AL(';H\_LDE\_USING\_THE\_INFOBOX\_STEPS',0)} [See related topics](#)

## Sizing a single control

1. Select the control.
2. Drag a handle in the direction you want to size the control.
3. Release the mouse button when the control is the desired size.

---

{button ,AL('H\_LDE\_SIZING\_SINGLE\_CONTROLS\_DETAILS',1)} [See details](#)

{button ,AL('H\_LDE\_SIZING\_A\_GROUP\_OF\_CONTROLS\_STEPS;H\_LDE\_USING\_THE\_GRID\_STEPS;H\_LDE\_SELECTING\_AND\_DESELECTING\_CONTROLS\_OVER',0)} [See related topics](#)

## Spacing controls

You can space multiple controls equally.

1. Select three or more controls.
2. Choose Dialog - Equal Spacing.
3. Choose how you want to space the controls:
  - "Vertical" spaces the controls equally between the top and bottom controls.
  - "Horizontal" spaces the controls equally between the left and right controls.

**Tip** After spacing a collection of controls, you can restore the controls to their original positions by choosing Edit - Undo.

---

{button ,AL(`;H\_LDE\_SELECTING\_AND\_DESELECTING\_CONTROLS\_OVER',0)} [See related topics](#)

## Starting the Dialog Editor

You can start the Dialog Editor from any Lotus product that supports it.

1. Start the Lotus product.
2. Open a document.
3. Choose Edit - Scripts & Macros (or equivalent), and choose Show Dialog Editor from the submenu.

**Note** You can display one new Dialog Editor window for each document.

---

{button ,AL(`;H\_LDE\_ENDING\_DIALOG\_EDITOR\_STEPS',0)} [See related topics](#)



**Details: Undoing and redoing actions**

Some actions cannot be undone. For example, you cannot undo the following actions:

- Changes you make to Toolbox setup
- Changes you make to SmartIcons setup
- Properties you set in the InfoBox
- Changes you make to the Dialog Editor preferences
- Changes in the front-to-back order of controls
- Changes to documents you have saved
- Any actions that occur while a dialog box is running

---

{button ,AL(`H\_LDE\_UNDOING\_AND\_REDOING\_ACTIONS\_STEPS',1)} [Go to procedure](#)

## Undoing and redoing actions

You can often use Undo or Redo to cancel or reinstate an action in the Dialog Editor.

To undo an action, choose Edit - Undo.



To redo an action, choose Edit - Redo.



**Note** You cannot undo some commands and actions.

---

{button ,AL(`H\_LDE\_UNDOING\_AND\_REDOING\_ACTIONS\_DETAILS',1)} [See details](#)

## Overview: Using constants

The Dialog Editor provides a set of constants that you can use in place of numeric arguments in certain LotusScript statements.

The constants used by the Dialog Editor are defined in LSDCNST.LSS. Use the %Include directive to incorporate this file in (Globals) or (Dialog Globals). The syntax for including this file is:

```
%Include "LSDCNST.LSS"
```

For example, if you have included LSDCNST.LSS in your script, you can display a dialog box as modal by using the following statement:

```
MyDialog.Show DIALOG_MODAL
```

---

{button ,AL(`;H\_LDE\_PROGRAMMING\_DIALOG\_BOXES\_OVER',0)} [See related topics](#)

## How do I use SmartIcons?

SmartIcons are buttons that represent mouse shortcuts for actions and commands. When you first load the Dialog Editor, a set of SmartIcons appears in a bar at the top of the window.



To use an icon, simply click it.

**Note** You can hide or show SmartIcons.

To tell you what an icon does, bubble Help appears by default when you position the mouse pointer over an icon. If bubble Help does not appear, position the mouse pointer on the icon and press the right mouse button. You can also hide or show SmartIcons bubble Help.



## Customizing SmartIcons

Use File - SmartIcons Setup to change the icons that appear in the bar, create additional sets of SmartIcons, and change the position of icons within each set.

For more information about customizing SmartIcons, see Help for the Lotus product in which you are using the Dialog Editor.

## Using the grid

You can display a grid and use it to align controls when you move or size them.

1. Choose File - Dialog Editor Preferences.



2. Select Show grid.
3. (Optional) Enter values for the width and height of the grid.

**Note** Values for width and height are set in Twips. The minimum value is 60 Twips and the maximum value is 480 Twips.

4. Select Align controls to grid.

---

```
{button ,AL(^;H_LDE_ADDING_A_CONTROL_STEPS;H_LDE_ADDING_A_CONTROL_USING_THE_CREATE_MENU_STEPS;H_LDE_ALIGNING_CONTROLS_STEPS;H_LDE_MOVING_CONTROLS_STEPS;H_LDE_SIZING_A_GROUP_OF_CONTROLS_STEPS;H_LDE_SIZING_SINGLE_CONTROLS_STEPS;H_LDE_SPACING_CONTROLS_STEPS',0)} See related topics
```

## Using the InfoBox

You can make changes to one control or several controls without closing the InfoBox.

1. Select what you want to modify.
2. Right-click the selection and choose Properties.
3. Click the tab for the properties you want to change.
4. Select one or more options.
5. (Optional) Move, collapse, or close the InfoBox.

---













{button ,AL(`;H\_LDE\_SETTING\_CONTROL\_PROPERTIES\_STEPS;H\_LDE\_SETTING\_DIALOG\_BOX\_PROPERTIES\_STEPS',0)} See related topics

## Overview: Working with dialog box controls

You use dialog box controls, such as text boxes, check boxes, option buttons, list boxes, and command buttons, to create a user interface. In addition to standard Lotus controls, you can also add third-party custom controls to dialog boxes you create in the Dialog Editor.

### Standard Lotus controls

The table below summarizes the standard Lotus controls included with the Dialog Editor.

Toolbox button	Class Name	Description
	LotusLabel	Displays text that users cannot (directly) change.
	LotusTextBox	Displays a box where users can enter text.
	LotusCheckBox	Displays a box where users can select or deselect options.
	LotusFrame	Groups a set of related controls, such as option buttons.
	LotusOptionButton	Lets users choose a (single) selection from one or more options.
	LotusListBox	Displays a list of items for users to choose from.
	LotusComboBox	Combines the features of a text box and a list box. Users choose an item in a combo box either by typing it or selecting from the list.
	LotusCommandButton	Performs an action.
	LotusImage	Displays bitmaps and other graphics.
	LotusSpinButton	Increments and decrements numbers. Also scrolls back and forth through a range of values. Must be used with another control that displays the values, such as a text box.
	LotusSlider	Displays a slider (rule), which users can use to select a value or range or values.
	LotusProgressBar	Indicates the (approximate) progress of an operation.

### Custom controls

You can also use any OCX custom control that is installed on your system. Custom controls provide specialized features or enhanced versions of the standard controls. To see a list of the custom controls registered on your

system, choose Files - Toolbox Setup from the Dialog Editor menu.

---

```
{button ,AL(';H_LDE_ADDING_A_CONTROL_STEPS;H_LDE_ADDING_A_CONTROL_USING_THE_CREATE_MENU_STEPS;H_LDE_ALIGNING_CONTROLS_STEPS;H_LDE_COPYING_CONTROLS_STEPS;H_LDE_DELETING_CONTROLS_STEPS;H_LDE_MOVING_CONTROLS_FRONT_AND_BACK_STEPS;H_LDE_MOVING_CONTROLS_STEPS;H_LDE_MOVING_CONTROLS_TO_OTHER_DIALOGS_STEPS;H_LDE_NAMING_A_CONTROL_STEPS;H_LDE_SETTING_CONTROL_PROPERTIES_STEPS;H_LDE_SIZING_A_GROUP_OF_CONTROLS_STEPS;H_LDE_SIZING_SINGLE_CONTROLS_STEPS;H_LDE_SPACING_CONTROLS_STEPS;H_LDC_CONTROLS_L$AL',0)} See related topics
```



