

The Lotus Dialog Editor provides an alternative to using the Microsoft Software Development Kit (SDK) to create dialog boxes that can be used in Lotus applications for Windows.

[Using Help](#)

[Using SmartIcons](#)

[Following Lotus Conventions and Standards](#)

[Tools for Creating a Dialog Box](#)

[Creating a Dialog Box](#)

[Modifying a Dialog Box](#)

[Adding Controls](#)

[Modifying Controls](#)

[Using Groups and Tabs](#)

[Control IDs and Values Returned by Controls](#)

[Saving a Dialog Box](#)

[Loading Applications from the Dialog Editor](#)

[Copying a Dialog Box to a Lotus Application](#)

[Using a Dialog Box in a Lotus Application](#)

The Dialog Editor provides online help for all functions.

To use help

1. Choose Help Contents.

The Dialog Editor displays a list of Help topics.

2. Choose a Help topic.

Keyboard: Press TAB to select the Help topic and press ENTER.

The Dialog Editor displays the Help topic.

See also:

[Using SmartIcons](#)

[Creating a Dialog Box](#)

SmartIcons are mouse shortcuts for commands. You can use SmartIcons in the Dialog Editor just as you use them in other Lotus products. You can display descriptions for the SmartIcons and specify the icons you want to display by modifying the SmartIcons.

To use SmartIcons

1. Choose Options SmartIcons.

The Dialog Editor displays its own set of SmartIcons across the top of the window.

2. Click an icon to implement the command.

To modify SmartIcons

1. Choose Options Modify SmartIcons.

The Dialog Editor displays the SmartIcons dialog box.

2. Select a position for the set of SmartIcons.
3. Modify the set of SmartIcons by adding, moving, removing, or grouping the icons.

See also:

[Creating a Dialog Box](#)
[Adding Controls](#)

Although you can be creative when designing dialog boxes, you should follow certain conventions and standards to give the dialog boxes the look and feel of Lotus dialog boxes.

Sizing conventions

Controls in Lotus dialog boxes adhere to the following standard sizes, which are measured in dialog units. A dialog unit varies with the resolution of your video display. The status box, however, displays the dimensions for each control in accurate dialog units. You can use the status box to determine whether a control's height and width follow the sizing conventions. You can size the control if it is incorrect.

<i>Control</i>	<i>Height</i>	<i>Width</i>
Static Text	8	*
Combo Boxes	13	*
Check Boxes	12	*
Radio Buttons	12	*
Edit Boxes	12	*
Push Buttons	14	40
Group Boxes	**	**
List Boxes	***	48

* As wide as the text, without truncating any text.

** Determined by the size of the items it surrounds.

*** As tall as can fit in the dialog box.

General standards

- Place push buttons in the top right corner of the dialog box, beginning with the OK (default) push button and followed immediately by the Cancel push button. Place any additional push buttons below the Cancel push button and vertically aligned with it.
- Position push buttons 2 dialog units apart.
- Make margins consistent in each box.
- Introduce edit, list, and combo boxes with static text followed by a colon.
- Create a group box for two or more check boxes or radio buttons.
- Avoid placing radio buttons and check boxes within the same list.
- Position radio buttons and check boxes vertically.
- Use 8-point MS Sans Serif font for all text in the dialog box.
- Capitalize the first letter of each word in push buttons and group boxes. For all other controls, capitalize only the first letter of the first word.
- Create shortcut keys (underlined letters) for all buttons, check boxes, and group boxes. Double-click the control and type an ampersand (&) immediately before the character in the Button Text edit box. For example, if you use the letter O in an Options push button as a shortcut key, the user can access that push button by holding ALT and pressing O (ALT+O).
- Do not create shortcut keys for the OK and Cancel push buttons, as the keyboard shortcuts for these two push buttons are ENTER and ESC, respectively.

See also:

[Creating a Dialog Box](#)

[Adding Controls](#)

[Modifying Controls](#)

Modifying a Dialog Box

Saving a Dialog Box

Using the Status Box

Using the Grid

Using Default Sizes

Tools for Creating a Dialog Box

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)



You can create a dialog box to allow the user to provide information while a macro or a script is running.

To create a dialog box

1. Choose File New.
2. Type the name for the dialog box.
3. If you want a title bar in the dialog box, type the title name.
4. Select the type of dialog box.

Plain

A dialog box without a title bar.

Title bar

A dialog box with a title bar. This allows a user to move the box by dragging the title bar to a new location.

Title bar with system menu

A dialog box with a title bar and a control menu. This allows a user to move or close the box using the keyboard.

5. If you want to display a Help button in the title bar, select Help icon.
The Help button does not display in the Dialog Editor, but it displays when the dialog box is used in the application.

Ami Pro: You must use a DlgKeyInterrupt function before a DialogBox function to make the Help button work correctly in a macro.

Improv: This feature is not supported by Improv Release 2.0.

1-2-3 Release 4: This feature is not supported by 1-2-3 Release 4.

6. Choose OK.
The mouse pointer changes to a cross.
7. Click the cross where you want to position the top left corner of the dialog box.
The Dialog Editor creates a dialog box.
8. Click the dialog box to select it.
The dialog box displays with black handles.
9. If you want to move the dialog box to another location in the window, drag the dialog box to the position.
10. If you want to size the dialog box, drag a handle until the dialog box is the size.

See also:

[Using SmartIcons](#)

[Modifying a Dialog Box](#)

[Adding Controls](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Saving a Dialog Box](#)

You can modify a dialog box by changing its name, title, and options. You can also modify the font used for all the text in a dialog box.

To modify a dialog box

1. Double-click any empty area in the dialog box or choose Edit Styles.
2. Specify a name and title.
3. Select a type of dialog box.

Plain

A dialog box without a title bar.

Title bar

A dialog box with a title bar. This allows a user to move the box by dragging the title bar to a new location.

Title bar with system menu

A dialog box with a title bar and a control menu box. This allows a user to move or close the box using the keyboard.

4. If you want to display a Help button in the title bar, select Help icon.

The Help button does not display in the Dialog Editor, but it displays when the dialog box is used in the application.

Ami Pro: You must use a DlgKeyInterrupt function before a DialogBox function to make the Help button work correctly in a macro.

Improv: This feature is not supported by Improv Release 2.0.

1-2-3 Release 4: This feature is not supported by 1-2-3 Release 4.

5. Choose OK.

To modify the font

1. Choose Edit Font.
The Dialog Editor displays the Select Font dialog box.
2. Select a face name.
3. Select a point size.
4. Choose OK.

Although the font does not display in the Dialog Editor, it displays when the dialog box is used in a macro or script.

See also:

[Using SmartIcons](#)

[Adding Controls](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Saving a Dialog Box](#)

[Following Lotus Conventions and Standards](#)

Control types

You can add the following types of controls to a dialog box:

[Push Button](#)

[Default Push Button](#)

[Radio Button](#)

[Check Box](#)

[Edit Box](#)

[List Box](#)

[Static Text](#)

[Combo Box](#)

[Group Box](#)

[Bitmap Button](#)

[Static Bitmap](#)

To add a control

1. Click the control icon or choose Control and choose the type of control you want to add to the dialog box.

The mouse pointer changes to a cross.

2. Click the cross where you want to position the top left corner of the control.

The Dialog Editor creates the control and automatically assigns it an ID.

3. Type the text for the control.

4. If you want to move the control to another location in the dialog box, drag the control to the position.

5. If you want to size the control, drag a handle until the control is the size you want or choose Edit Size Item and specify the dimensions.

You can modify any control by changing its style and size. You can modify a button or static text by placing a picture in it instead of text.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Modifying a Dialog Box](#)

[Saving a Dialog Box](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Following Lotus Conventions and Standards](#)



You can modify any control by changing its ID, style, or size.

To modify the style of a control

1. Double-click the control or select the control and choose Edit Styles.
2. Specify the options.

Button, check box, or group box: Specify the ID, Button Text, and Button Type. If you want to use a picture instead of text, select Bitmap as the Button Type and specify a file name instead of text in the Button Text edit box. If you want to create a shortcut key for the control, type an ampersand (&) immediately before a character in the Button Text edit box. That character appears underlined in the dialog box and acts as a keyboard shortcut for the control.

Edit box: Specify an ID and whether you want to implement Multiple lines (the user can press CTRL+ENTER to create multiple lines in the box) and Auto scroll (the edit box automatically scrolls to display additional text).

List box: Specify an ID and whether you want to implement Multiple selection (the user can select more than one item in the list box) and Sort (the items in the list box are arranged in alphanumeric order). **Note** Sort is not supported by 1-2-3 Release 4.

Combo box: Specify an ID, Style, and whether you want to implement a Vertical scroll bar and Sort.

Static text or static bitmap: Specify an ID, Text, and Style for the text or picture.

3. Choose OK.

To modify the size of a control

1. Select the control and choose Edit Size Item.
2. Specify an ID.
3. Specify the dimensions using the coordinate measurements displayed in the status box.
The unit of measurement is a dialog unit, which is the measurement used in the status box.
X
The horizontal starting position of the control.
Y
The vertical starting position of the control.
CX
The width of the control.
CY
The height of the control.
4. Choose OK.

See also:

[Using SmartIcons](#)

[Modifying a Dialog Box](#)

[Saving a Dialog Box](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can create push buttons to allow the user to close the dialog box or display another dialog box.

The default push button should always be OK and use ID number 1, and the Cancel push button should always use ID number 2. Since the Dialog Editor automatically numbers the controls, you should create the default push button first, and then create the Cancel push button. Both OK and Cancel buttons close the dialog box; OK returns the value 1, and Cancel returns the value 0.

Other push buttons that close a dialog box use ID numbers 3 through 19 and return ID values. Push buttons that do not close a dialog box use ID numbers 20 through 99 and return ID values through a product specific callback function.

You can modify a push button by changing its ID or text, or by using a picture instead of text.

Related SmartIcons



Create a default push button

See also:

- [Using SmartIcons](#)
- [Modifying Controls](#)
- [Modifying a Dialog Box](#)
- [Using Radio Buttons](#)
- [Using Check Boxes](#)
- [Using Edit Boxes](#)
- [Using List Boxes](#)
- [Using Static Text](#)
- [Using Combo Boxes](#)
- [Using Group Boxes](#)
- [Using Bitmap Buttons](#)
- [Using Static Bitmaps](#)
- [Using Groups and Tabs](#)
- [Using the Status Box](#)
- [Using the Grid](#)
- [Using Default Sizes](#)
- [Control IDs and Values Returned by Controls](#)
- [Following Lotus Conventions and Standards](#)



You can create radio buttons to allow the user to select options that are mutually exclusive. The user can select only one radio button at a time, as on a car radio. When you push one button, the previously selected button is deselected. Only one radio button can be active at a time.

Radio buttons use ID numbers 20 through 99 and return Boolean (TRUE (1) or FALSE (0)) values, with only the active button returning a TRUE value. The common method of determining how to branch the macro or script, given the values of the radio buttons, is to use a series of IF-THEN-ELSE, SWITCH CASE, or SELECT CASE statements, where each variable set for a radio button is evaluated to see if it is true.

You can modify a radio button by changing its ID or text.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can create check boxes to allow the user to select options that are not mutually exclusive. Check boxes use ID numbers 20 through 99 and return Boolean (TRUE (1) or FALSE (0)) values. You can modify a check box by changing its ID or text.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can create and size edit boxes to allow the user to enter string data.

Unlike radio buttons and check boxes, edit boxes do not have any text attached to them. You should create static text followed by a : (colon) to label each edit box.

Edit boxes use ID numbers 8000 through 8999. The ID numbers fill edit boxes with data using product-specific functions. They return string values through product-specific functions.

In Lotus applications other than 1-2-3 Release 4, you can automatically fill an edit box with selected data from a list box that uses an ID number exactly 1000 greater than the ID number for the edit box.

You can modify an edit box by changing its ID or by selecting options.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can use list boxes to create a list of data in a box. The user can then select one or more items in the list box.

List boxes use ID numbers 9000 through 9499. In Lotus applications other than 1-2-3 Release 4, a list box with an ID number of 9001 fills with the contents of the current directory.

Other list boxes fill with data using product-specific functions:

Ami Pro: Returns string values for selected items in a list box.

Improv: Returns string values for selected items in a list box.

1-2-3 Release 4: Returns the index number for a single selected item; in the case of multiple selections, 1-2-3 returns the total number of selected items and identifies which items are selected.

You can use selected data in a list box to automatically fill an edit box that uses an ID number exactly 1000 lower than the ID number for the list box.

You can modify a list box by changing its ID or by selecting options.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can use static text to give the user information, such as a prompt for the kind of information that should be typed in an edit box. You can also use static text to give the user instructions or information specific to the macro or script.

Static text controls use ID numbers 1000 through 7999 and return no value.

In Lotus applications other than 1-2-3 Release 4, a static text control with an ID number of 7999 fills with the current path if the dialog box contains a list box with an ID number of 9001. This is useful for providing the current path when the user needs to select a file name in a list box.

You can modify static text by changing its ID or text, or by using a picture instead of text.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



Combo boxes combine a list box with an edit box. You can use combo boxes to allow the user to select an item in a list box and then display that item in the edit box.

Combo boxes use ID numbers 9500 through 9999 and fill with data through product specific functions. They return string values for selected items in a list box.

In Lotus applications other than 1-2-3 Release 4, you can modify a combo box by changing its ID or style, or by selecting options. If you do not implement sort for a combo box, the first item in the list box appears in the edit box portion of the combo box as the default and all other items appear in the order in which they were placed in the list box. If you implement sort, the item with the lowest alphanumeric value appears in the edit box portion of the combo box.

In 1-2-3 Release 4, the edit portion of the combo box is always blank when the dialog box first appears.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can use group boxes to separate one section of a dialog box from other sections. This makes the dialog box easier for the user to understand.

The group box has static text attached to it and a static line drawn around it. You can create a group box over controls to visually separate those controls from other controls in the dialog box.

Group boxes use ID numbers 20 through 29 and return no value.

You can modify a check box by changing its ID or text.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Bitmap Buttons](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can create push buttons with images on them; like other push buttons, bitmap buttons allow the user to close the dialog box or display another dialog box.

When you choose Bitmap Button as the control type, enter the file path and file name of the bitmap you want to appear on the push button.

Bitmap buttons can have the same ID numbers as push buttons.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Modifying a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Static Bitmaps](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



Static bitmaps appear in the dialog box as static images.

When you choose Static Bitmap as the control type, enter the file path and file name of the bitmap you want to appear in the dialog box.

Static bitmaps can have ID numbers from 1000 through 7999.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Modifying a Dialog Box](#)

[Using Push Buttons and Default Push Buttons](#)

[Using Radio Buttons](#)

[Using Check Boxes](#)

[Using Edit Boxes](#)

[Using List Boxes](#)

[Using Static Text](#)

[Using Combo Boxes](#)

[Using Group Boxes](#)

[Using Bitmap Buttons](#)

[Using Groups and Tabs](#)

[Using the Status Box](#)

[Using the Grid](#)

[Using Default Sizes](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)



You can create and modify groups and tabs to allow the user to move around in a dialog box using the keyboard. When you create a group, the user can press the arrow keys to access any controls in the group. When you create a tab, the user can press TAB to move from one group to another.

Creating groups also allows you to separate mutually exclusive items, such as radio buttons, from those items that are not mutually exclusive, such as check boxes.

Before you create groups, you must specify the order in which you want the controls to be accessed using the keyboard.

To create groups

1. Choose Edit Order Items.

The Dialog Editor displays the Order Groups dialog box, which lists the controls and their IDs in the order in which you created them.

An asterisk (*) appears before any control that begins a group. A greater than symbol (>) appears before any control that acts as a tab.

2. If you need to arrange the controls so that they are in the correct order for a group, select a control and move it up or down to the position.

This does not change the position of the controls in the dialog box. It only determines the order in which the controls can be accessed using the keyboard.

3. Select the first control you want in a group.

4. Choose Add Group.

The Dialog Editor includes all controls below the one you selected in the group.

5. Repeat steps 3 and 4 to create additional groups.
6. Choose OK.

To create tabs

1. Complete the steps for creating groups.
2. Select the control you want to make accessible by pressing TAB. This is usually the first control in a group.
3. Choose Add Tab.
4. Repeat steps 2 and 3 to create a tab for each group.
5. Choose OK.

To modify groups and tabs

You can modify groups and tabs. The Add Group and Add Tab command buttons change to Delete Group and Delete Tab when you select a control that begins a group or acts as a tab.

1. Choose Edit Order Items.
2. Select a control you want to modify.
3. Choose the command button.

Delete Group

The control no longer begins a group. The Dialog Editor removes all controls from that group.

Delete Tab

The control no longer acts as a tab.

4. Choose OK.

See also:

Using SmartIcons
Modifying Controls
Saving a Dialog Box
Using Radio Buttons
Using Check Boxes
Using Group Boxes
Using the Status Box
Using the Grid
Using Default Sizes
Control IDs and Values Returned by Controls
Following Lotus Conventions and Standards

You can use the status box to display the position, width, height, and ID for either the dialog box or a selected control. You can move the status box window anywhere on the screen.

To use the status box

1. Choose Options Status Box.

The Dialog Editor displays the status box.

2. To close the status box, double-click the control menu box in the status box or choose Options Status Box.

See also:

[Using SmartIcons](#)

[Using the Grid](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Following Lotus Conventions and Standards](#)



You can use the grid to align controls. When you implement the grid, all controls "snap to" the grid intersections when you create or move them.

To use the grid

1. Choose Options Grid.

The Dialog Editor displays the Grid Settings dialog box.

2. Specify the minimum horizontal and vertical settings. The normal setting is 2.

The unit of measurement is a dialog unit, which is the measurement used in the status box.

The horizontal and vertical settings determine the distance between the intersections in the grid and allow you to control the alignment precision for the controls.

3. Choose OK.

See also:

[Using SmartIcons](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Following Lotus Conventions and Standards](#)



You can use Default Sizes to set the default sizes for push buttons, edit boxes, list boxes, radio buttons, check boxes, and static text. The Dialog Editor uses the sizes you specify whenever you create these controls.

To use default sizes

1. Choose Options Default Sizes.

The Dialog Editor displays the Default Sizes dialog box.

2. Specify the width and height for each control.

The unit of measurement is a dialog unit, which is the measurement used in the status box.

3. Choose OK.

See also:

[Using SmartIcons](#)

[Modifying a Dialog Box](#)

[Adding Controls](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Following Lotus Conventions and Standards](#)

When you create a control in a dialog box, the Dialog Editor automatically assigns an ID number to the control. When you use the dialog box in a macro or script, the controls return the following values.

<i>Control</i>	<i>ID</i>	<i>Return Value</i>
Default Push Button	1	1; closes the dialog box
Cancel Push Button	2	0; closes the dialog box
Push Buttons	3 thru 19	ID value; closes the dialog box
Bitmap Buttons	3 thru 20	ID value*
Push Buttons	20 thru 99	ID value; through a callback function*
Radio Buttons	20 thru 99	Boolean value
Check Boxes	20 thru 99	Boolean value
Group Boxes	20 thru 99	None
Static Text	1000 thru 7999	None
Static Bitmap	1000 thru 7999	None
Special Static Text	7999	None; displays the current path**
Edit Boxes	8000 thru 8999	String value
List Boxes	9000 thru 9499	String value
Special List Box	9001	String value**
Combo Boxes	9500 thru 9999	String value

* In 1-2-3 Release 4, all buttons close the dialog box.

** Not in 1-2-3.

See also:

[Adding Controls](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Following Lotus Conventions and Standards](#)

[Loading Applications from the Dialog Editor](#)



You can save a dialog box as an ASCII file by choosing File Save and specifying a file name. The Dialog Editor places all the formatting information for the dialog box into an ASCII file with a .DLG extension.

If you need to edit a dialog box, choose File Open in the Dialog Editor and specify the name of the file. You can then modify the dialog box or the controls and save the changes by choosing File Save.

The ASCII file is formatted in a certain way:

- The first line contains the word DIALOG and the name of the dialog box.
- The next line is the description of the dialog box itself. This line contains six numbers and three strings.
- The first number represents the options that have been set for the dialog box, such as its type (plain, title bar, or title bar with system menu). The second number is an integer which is the total number of controls in the dialog box. The third number is the horizontal starting position of the top left corner of the box (x). The fourth number is the box's vertical starting position (y). The fifth and sixth numbers are the horizontal (cx) and vertical (cy) widths.
- The first two string values are NULL (empty) and the last item is the title of the dialog box (in quotes).
- If the font is changed, the next line describes the font used in the dialog box. The preferred font (to match the Lotus dialog boxes) is 8-point MS Sans Serif.
- The remaining lines describe the controls in the dialog box. Each line contains the following information, in this order: the control's horizontal starting position (x), vertical starting position (y), width of control (cx), height of control (cy), the control's ID number, the control's options and attributes, the type of control (in quotes), the description of the control (which the user sees, also in quotes), and a zero.
- The last line of the dialog box description contains only the words END DIALOG.

Example of a dialog box ASCII file

```
DIALOG ExampleBox
-2134376448 4 106 78 160 42 "" "" "Sample Dialog Box"
FONT 8 "Helv"
6 14 34 8 1000 1342308352 "static" "&Filename:" 0
46 14 56 12 8000 1350631552 "edit" "" 0
116 4 40 14 1 1342373889 "button" "OK" 0
116 22 40 14 2 1342373888 "button" "Cancel" 0
END DIALOG
```

See also:

[Creating a Dialog Box](#)

[Adding Controls](#)

[Modifying Controls](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)

[Loading Applications from the Dialog Editor](#)

You can load Ami Pro, Improv, or 1-2-3 Release 4 from the Dialog Editor, or switch to the application if it is already running. This enables you to copy and paste dialog boxes. These options work only if the application you wish to load is in your DOS path or is already running.

To load or switch to Ami Pro

Choose Options Ami Pro.

The Dialog Editor loads Ami Pro. You can create or open a macro (.SMM) file.

To load or switch to Improv

Choose Options Improv.

The Dialog Editor loads Improv. You can create or open a script.

To load or switch to 1-2-3 Release 4

Choose Options 123.

The Dialog Editor loads 1-2-3. You can paste the dialog box into a worksheet, where it appears as a dialog-description table.

See also:

[Using SmartIcons](#)

[Saving a Dialog Box](#)

[Copying a Dialog Box to a Lotus Application](#)

[Using a Dialog Box in a Lotus Application](#)

Copying a Dialog Box to a Lotus Application

For more information, select an application icon.



Ami Pro



Improv



1-2-3 Release 4

Using a Dialog Box in a Lotus Application

For more information, select an application icon.



Ami Pro



Improv



1-2-3 Release 4



You can copy a dialog box and paste it into an Ami Pro macro (.SMM) file. All the formatting information for the dialog box is then placed into that file. This allows you to access a dialog box from within a macro and to combine dialog boxes into one file.

If you want, you can delete the .DLG file after you place the formatting information into a macro.

You can also copy the formatting information for a dialog box from an ASCII or macro file and paste it into the Dialog Editor. You can then create a new dialog box from an existing one.

To copy a dialog box into an Ami Pro macro file

1. In the Dialog Editor, move the mouse pointer outside the dialog box and either click the mouse or press SPACEBAR to deselect the dialog box.

2. Choose Edit Copy.

The dialog box formatting information is placed on the Clipboard.

3. In Ami Pro, open the macro file into which you want to paste the dialog box formatting information.

4. Place the insertion point above or below any FUNCTION/END FUNCTION statement.

5. Choose Edit Paste.

The formatting information for the dialog box appears in the file.

To copy a dialog box into the Dialog Editor

1. In the macro, select all the dialog box formatting information, starting with the DIALOG line and including the END DIALOG line.

2. Choose Edit Copy.

3. In the Dialog Editor, choose Edit Paste.

The dialog box appears in the Dialog Editor.

See also:

[Modifying a Dialog Box](#)

[Modifying Controls](#)

[Saving a Dialog Box](#)

[Using a Dialog Box in Ami Pro](#)

[Control IDs and Values Returned by Controls](#)

[Following Lotus Conventions and Standards](#)

[Loading Applications from the Dialog Editor](#)

After you paste the formatting information for a dialog box into an Ami Pro macro file, you can use the DialogBox function to display the dialog box.

You can also use several other Ami Pro macro functions such as

[DlgKeyInterrupt](#), [FillEdit](#), [FillList](#), [GetDialogField\\$](#), [GetDlgItem](#), [GetDlgItemText](#), [SetDlgCallBack](#), [SetDlgItemText](#) to enhance the dialog box.

Some of these functions are used in the following example:

- The [FillEdit](#) function fills the edit box (ID 8000) with the text "FILENAME.EXE".
- If a user presses F1, the [DlgKeyInterrupt](#) calls the HelpMac function which displays the text "No Help" in a message box.
- The [DialogBox](#) function displays the [ExampleBox dialog box](#) and returns a value to SampleBox. If that value is -1 or 0, the macro exits.
- The [GetDialogField\\$](#) function reads the text in the edit box (ID 8000) and returns this value to Filename. A message box displays with the returned Filename.

Example of using a dialog box in an Ami Pro macro

```
FUNCTION CallBox()
FillEdit(8000, "FILENAME.EXE")
DlgKeyInterrupt ([F1],"HelpMac")
SampleBox = DialogBox(".", "ExampleBox")
IF SampleBox = -1 'If we couldn't find the box file,
    EXIT FUNCTION 'Exit the macro.
ELSEIF SampleBox = 0 'Did the user cancel the box?
    EXIT FUNCTION 'Yes, exit the function
ENDIF
Filename = GetDialogField$(8000)
Message("The Filename is {Filename}")
END FUNCTION
```

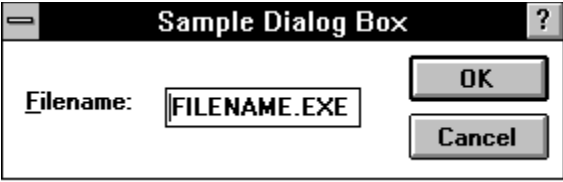
```
FUNCTION HelpMac()
Message("No Help")
END FUNCTION
```

```
DIALOG ExampleBox
-2134376448 4 106 78 160 42 "" "AmiDialog" "Sample Dialog Box"
FONT 8 "Helv"
6 14 34 8 1000 1342308352 "static" "&Filename:" 0
46 14 56 12 8000 1350631552 "edit" "" 0
116 4 40 14 1 1342373889 "button" "OK" 0
116 22 40 14 2 1342373888 "button" "Cancel" 0
END DIALOG
```

See also:

[Modifying a Dialog Box](#)
[Modifying Controls](#)

Copying a Dialog Box to Ami Pro
Control IDs and Values Returned by Controls
Following Lotus Conventions and Standards
Loading Applications from the Dialog Editor



You can use a dialog box in Improv by creating a script. For help on using dialog boxes in scripts, see "Dialog Box Access" in the LotusScript Help.

See also:

[Lotus Dialog Editor Contents](#)



To copy a dialog box to a worksheet file

1. In the Dialog Editor, deselect all controls by clicking the mouse outside the dialog box.
2. Choose Edit Copy.
The Dialog Editor places the dialog-box information on the Clipboard.
3. In 1-2-3 Release 4, open the worksheet file into which you want to paste the dialog-box information.
4. Move the cell pointer to the cell you want to be the first cell of the dialog-description table.
The dialog-description range requires 11 columns and as many rows as there are dialog box controls, plus two rows. 1-2-3 also requires that the twelfth column be blank.
5. Choose Edit Paste.
1-2-3 enters the dialog-box information in the dialog-description table.

To copy a dialog box back to the Dialog Editor

You can make changes to a dialog box by copying the dialog-description table from 1-2-3 to the Dialog Editor. You can then edit the dialog box and copy it back to 1-2-3 as described above.

1. In 1-2-3, select the entire range that contains the dialog description table.
2. Choose Edit Copy.
3. In the Dialog Editor, choose Edit Paste.
The dialog box appears.

See also:

[Lotus Dialog Editor Contents](#)

After you create a dialog box in the Dialog Editor and copy it to the Clipboard, you paste it in a worksheet file so that you can use it in a macro.

In the file, the dialog box takes the form of a dialog-description table. This table contains 11 columns and as many rows as there are dialog-box controls, plus two rows. The two rows contain labels that mark the beginning and end of the table.

1-2-3 uses the first nine columns of the table to store information about the contents and layout of the dialog-box. The table requires two more columns, an input column and an output column.

Note 1-2-3 requires that the "twelfth" column--that is, the first column to the right of the output column--be blank. This column is not, however, part of the dialog-description table.

To see a sample dialog-description table, start 1-2-3 and open the file UIMACROS.WK4; then click the worksheet tab {Dialog}, or press CTRL+PG UP ten times to go to worksheet K.

See also:

[Copying a Dialog Box to 1-2-3 Release 4](#)
[Lotus Dialog Editor Contents](#)

