

Using the Dialog Editor

This chapter describes how to use the **Dialog Editor** to interactively design dialog boxes to be used your **DCL** scripts. A mouse is required to use the **Dialog Editor**.

Starting the Dialog Editor

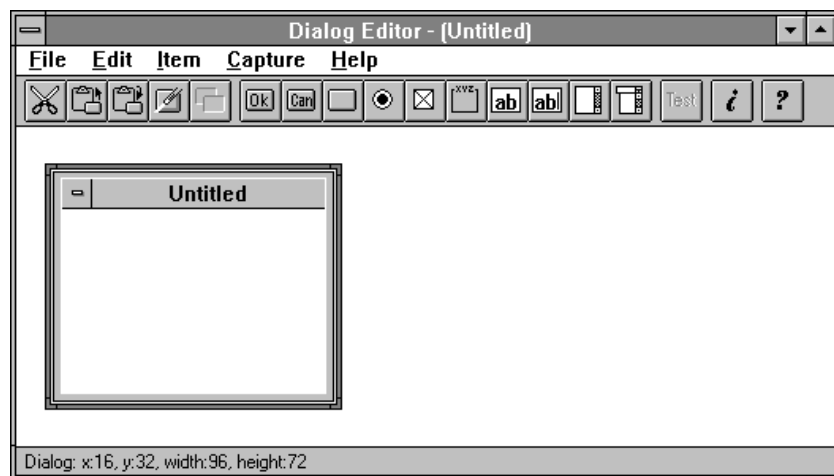
The procedure for starting the **Dialog Editor** differs, depending upon whether you are creating a new dialog box or modifying an existing one.



To start the dialog editor to create a new dialog box template

1. Position the insertion point where you want to include the new dialog box template.
2. Choose Dialog Editor from the Tools menu.

The **Dialog Editor** window containing a blank dialog box template is displayed.

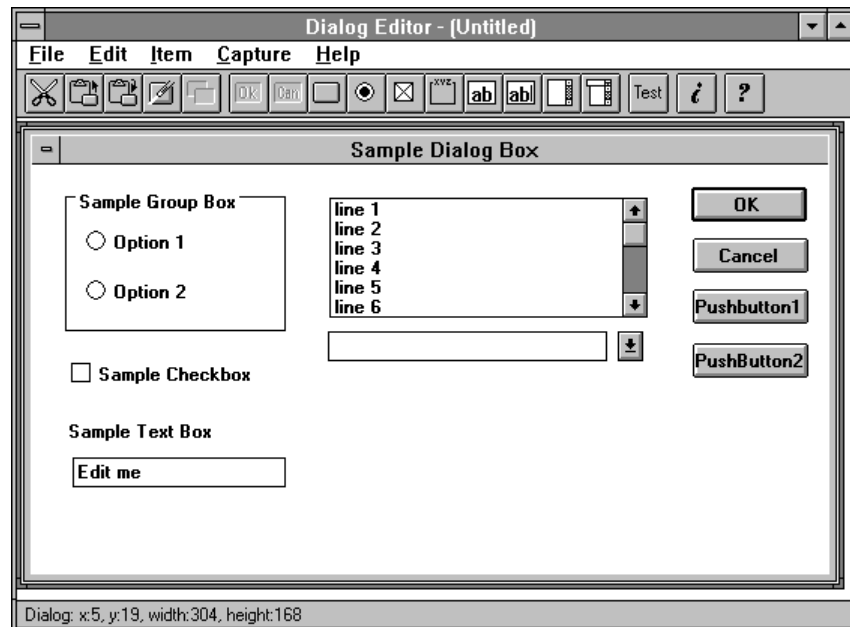




To start the dialog editor to modify an existing dialog box template

1. In the script, select the block of text containing the dialog box template. The block of text begins with a `Begin Dialog` statement and ends with an `End Dialog` statement. Be sure to include the `Begin Dialog` and `End Dialog` statements in the block of selected text.
2. Choose Dialog Editor from the Tools menu.

The text template is translated into a graphical representation in the **Dialog Editor** window.



Creating or Modifying a Dialog Box Template

The general process of creating or modifying a dialog box is:

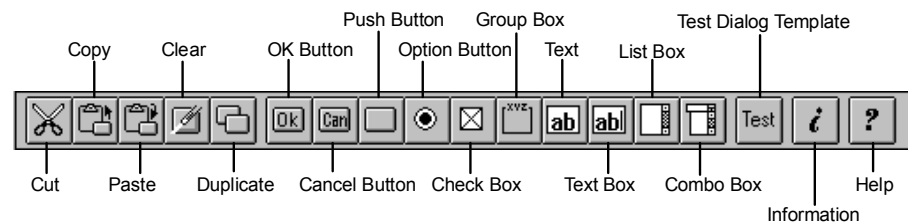
- Start the **Dialog Editor** as described under “Starting the Dialog Editor” in this chapter. If you are creating a new dialog box template, an empty dialog box template is displayed. If you are modifying an existing dialog box template, the dialog box template contains the currently defined controls.

- Size the dialog box template by dragging the edges and corners.
- Add dialog box controls (buttons, options, boxes, text, etc.) to the dialog box template as described under “Adding Dialog Box Controls” in this chapter.
- Define the dialog box controls as described under “Defining Dialog Box Controls” in this chapter.
- Test the dialog box template as described under “Testing the Dialog Box Template” in this chapter.
 - When you are satisfied with the dialog box template, choose Exit & Update from the File menu. The **DCL** statements defining the dialog box template are inserted in your script.

Note: To see a dialog box definition used in a script, see “Sample Script” in this chapter.

Using the Toolbar

The Toolbar is a row of buttons representing frequently used **Dialog Editor** commands. Instead of using the menus, you can choose a Toolbar button to execute a command.



The Dialog Editor Toolbar

Using the Status Bar

The Status Bar appears at the bottom of the **Dialog Editor** window. It provides the name, coordinates, and size of the selected control.

PushButton: x:45, y:41, width:41, height:14

The Dialog Editor Status Bar

Adding Dialog Box Controls

You can add the following dialog box controls: OK button, Cancel button, Push button, Option (radio) button, Check box, Group box, Text, Text box, List box, Combo box.



To add a dialog box control

1. Select the control from the Item menu or Toolbar. This causes the cursor to take a shape resembling the control.
2. Move the cursor to the desired location in the dialog box template and click the left mouse button.
3. Size the control (by dragging edges and corners).

For all controls except the OK and Cancel buttons, you need to define the control as described under “Defining Dialog Box Controls” in this chapter.

To see a dialog box definition used in a script, see “Sample Script” in this chapter.

Defining Dialog Box Controls

After adding a dialog box control, you need to provide specific information to define the control. For example after adding a Push button, you need to define the label to appear on the button.



To define a dialog box control

1. Select the control by clicking on it, using the left mouse button. A border appears around the selected control. To select the entire dialog box, click the title bar.
2. Choose Info from the Edit menu or Toolbar.

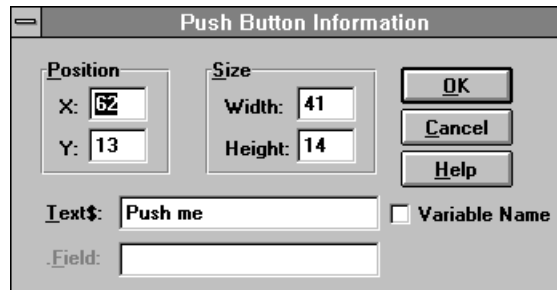
OR

Click on the control again, this time using the right mouse button.

OR

Combine Steps 1 and 2 by double-clicking on the control.

The <Dialog Box Control> Information dialog box is displayed. The title and contents of this dialog box vary depending on the control you are defining.



3. Use the Position and Size group boxes to change the upper left coordinates and size of the control if desired. (For an explanation of the unit of measurement used in dialog boxes, see the `Begin Dialog...End Dialog` command in Chapter 8, “Command Reference.”)
4. Complete the Text\$, Array\$, and .Field items, depending on the type of control you are defining. For more information see, “Using Text\$, Array\$, and .Field Items” in this chapter.
5. Choose OK.

Using Text\$, Array\$, and .Field Items

This topic summarizes how to use the Text\$, Array\$, and .Field items in the <Dialog Box Control> Information dialog box to define different dialog box controls. Each of these item is related to a **DCL** command. For an example of these variables in a **DCL** script, see “Sample Script” in this chapter.

For information about the **DCL** commands in the following paragraphs, see Chapter 8, “Command Reference.”

Text\$ Note: If the text of a label is to be read from a string variable, check the Variable Name box on. In the **Dialog Editor** window, the name of the variable is displayed rather than literal text.

The Dialog Box

Use the Text\$ field to enter a title for the dialog box.

Provide a Name for **DCL** to use to reference this dialog box template.

Related Command:

- `Begin Dialog`

OK and Cancel button

No action required. The `Dialog()` function, which displays the dialog in a running script, returns a value of -1 if the OK button was pressed or 0 for the Cancel button.

Related Commands:

- `OKButton`
- `CancelButton`

Push button

Use the `Text$` field to enter the label for the button. The `Dialog()` function returns a value greater than 0 if a Push button was pressed. 1 is the value of the first Push button in the dialog box template, 2 is the second, etc.

Related Command:

- `PushButton`

Check box

Use the `Text$` field to enter the label for the check box.

Provide a `.Field` variable name for this check box. It contains a -1 if the box is checked or a 0 if it is blank. This variable name must be unique within the dialog box template.

Related Command:

- `CheckBox`

Option group and buttons

An Option group is a collection of related Option buttons. For each Option button, use the `Text$` field to enter the label for the option.

Use the `.Field` item to enter the same variable name for each Option button in the group. This integer variable indicates which option in the group is selected. A value of -1 indicates that no option is selected. 0 indicates that the first option in the dialog box template is selected, 1 indicates the second, etc.

Related Commands:

- `OptionGroup`
- `OptionButton`

List box

Use the `Array$` item to provide the name of an array of strings to be displayed in the list box.

Use the `.Field` item to provide the name of an integer variable, which serves as the index to the array. This variable points to the selected array element. The first array element is 0, the second is 1, etc. When the array index is set to -1, no item is selected.

Related Command:

- `ListBox`

Combo box

Use the `Array$` item to provide the name of an array of strings to be displayed in the list box.

Use the `.Field` item to provide the name of a variable that contains the string that occupies the combo box.

Related Command:

- `ComboBox`

Text

Use the `Text$` field to enter the text to be displayed.

Related Command:

- `Text`

Text box

Provide a `Field` variable name for this text box. This string variable will contain the text for the text box. This variable name must be unique within the dialog box template.

Related Command:

- `TextBox`

Group box

Use the `Text$` field to enter the label for the group.

Related Command:

- `GroupBox`

Cutting, Copying, and Pasting Controls

The **DCL Dialog Editor** lets you perform various graphical “editing” tasks.

Use the Cut command on the Edit menu to remove a control you have selected from the dialog box template and store it in the Windows clipboard.

Use the Copy command to copy the selected control to the clipboard.

Use the Paste command to insert the contents of the clipboard in the dialog box template.

Use the Clear command to remove the selected control from the dialog box template.

Use the Duplicate command to make a copy of the selected control in the dialog box template.

Testing the Dialog Box Template

To test the dialog box template, choose Test Dialog from the File menu. Press F2 to return to the **Dialog Editor**.

Capturing Dialog Boxes

You can capture the controls from another dialog box to the Windows clipboard and then paste those controls in the **Dialog Editor** dialog box template.



To capture the controls from another dialog box

1. If the dialog box you want to capture is visible with the **Dialog Editor** window displayed, choose the Capture in Place command from the Capture menu.

If the dialog box is hidden by the **Dialog Editor** window, choose the Capture from Back command. If you choose this command, the **Dialog Editor** window is hidden while you capture the dialog box.

The mouse cursor takes the shape of a hook. When the cursor is over a capturable dialog, it looks like this:



2. Click on the dialog box you want to capture.

The dialog box controls are stored in the Windows clipboard as **DCL** commands.

Return to the **Dialog Editor** window.

3. Click inside the dialog box template in the **Dialog Editor** window.
4. Choose the Paste command from the Copy menu.

Opening an Existing File

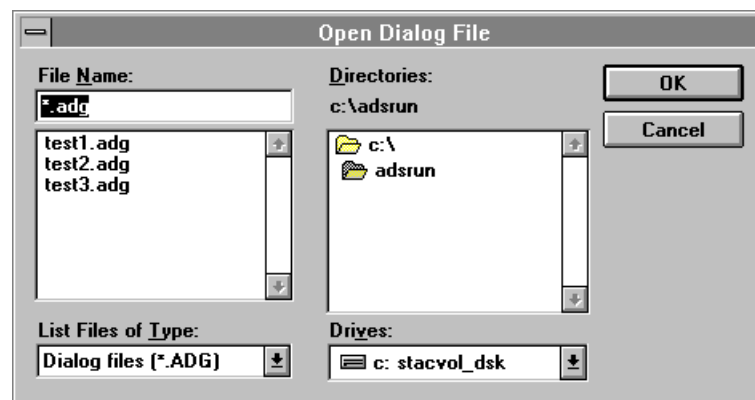
You can save dialog box templates in text files. When you open the text file, the **Dialog Editor** translates the **DCL** commands into a graphical representation of the dialog box template.



To open an existing file

1. Choose the Open command from the File menu.

The Open Dialog File dialog box is displayed.



2. Select the Drive and Directory containing the file.
3. Enter the File Name. Or select the file from the File Name list box.

You can change the types of files displayed in the File Name list box by selecting another file type from the List Files of Type box.

4. Choose OK.

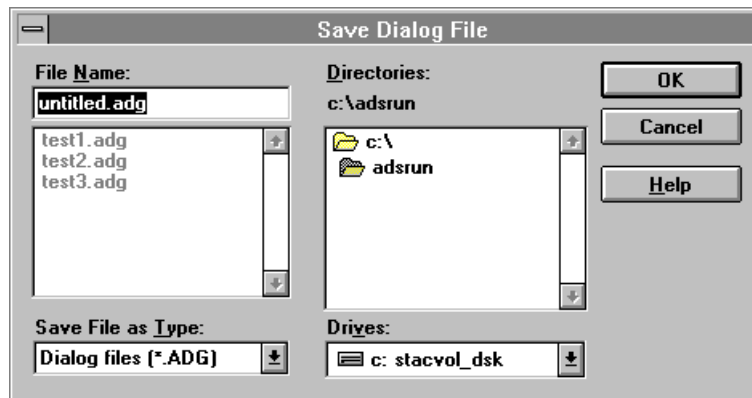
Saving a New Dialog Box Template

You can save the dialog box template to a text file. The **Dialog Editor** translates the graphical representation of the dialog box template into **DCL** commands.



To save a new dialog box template

1. Choose the Save As command from the File menu.



2. If you need to change the current path, select the Drive and Directory to contain the file.
3. Enter the File Name.
4. Choose OK.

Saving an Existing File

To save an existing file, choose the Save command from the File menu.

Saving an Existing File Under a New Name

Use the Save As command to save an existing file under a new name.



To save an existing file under a new name

1. Choose the Save As command from the File menu.
The Save Dialog File dialog box is displayed.
2. Select the Drive and Directory to contain the file.
3. Enter the File Name.
4. From the Save File As Type box, select the desired file type.
5. Choose OK.

If a file having the same name already exists, you will be asked to confirm that you want to overwrite the file.

Running the Dialog Editor Standalone

You can also run the **Dialog Editor** as a standalone application, not as a part of the **DCL Editor**. When you run the **Dialog Editor** in this way, you must save your dialog box template as an .ADG file.

To start the **Dialog Editor** standalone, use the File/Run command in **Applications Manager** (or Program Manager) to start the program ADSDLG11.EXE in the NETTOOLS directory. Alternatively you can use File Manager to start the program.

Getting Help

DCL provides extensive Help that describes how to use the graphical dialog box building environment.

To display the Table of Contents for the **Dialog Editor** Help, choose Contents from the Help menu or press F1.

To search an index of **DCL** Help keywords, choose Search for Help On from the Help menu.

For information on Microsoft Windows Help, choose How to Use Help from the Help menu.

Context Sensitive Help

When you request context-sensitive Help, the **Dialog Editor** displays a particular Help topic, based on your current activity. Context-sensitive Help is provided for each menu item and Toolbar button.



For help on a menu item (command)

1. Press Shift + F1.

The cursor changes to indicate that you are requesting context-sensitive Help.

2. Choose the menu and command you want Help on.



For help on a toolbar button

1. Press Shift + F1.

The cursor changes to indicate that you are requesting context-sensitive Help.

2. Choose the Toolbar button you want Help on.

Sample Script

The following **DCL** script illustrates the definition and display of a dialog box as well as the interpretation of the user's input. For information about any **DCL** commands, see Chapter 8, "Command Reference."

The main programming tasks required to implement a dialog box are:

- Initialization
- Defining the dialog box template
- Displaying the dialog box
- Interpreting the user's interaction with the dialog box

The ← symbol indicates that the command should continue on the same line.

```

sub main()
    'Example showing the entire process of defining, displaying,
    'and interpreting the input from a dialog box.

    'Initialization
    '-----
    'ListBox$ and ComboBox are single-dimensional arrays to hold the
    'contents of the list and combo box in the dialog box.
    Dim ListBox1$() as string
    Dim ComboBox1$() as string

    'Define dialog box
    '-----
    Begin Dialog UserDialog 16,32,304,168, "Sample Dialog Box"
        OKButton 251,9,44,14
        CancelButton 252,30,44,14
        PushButton 252,51,44,14, "Pushbutton1"
        PushButton 252,73,44,14, "PushButton2"
        GroupBox 13,9,84,59, "Sample Group Box"
        OptionGroup .OptionGroup1
            OptionButton 21,24,65,14, "Option 1"
            OptionButton 21,44,66,14, "Option 2"
        CheckBox 15,78,79,14, "Sample Checkbox", .CheckBox1
        Text 14,105,79,8, "Sample Text Box"
        TextBox 16,120,81,12, .TextBox1
        ListBox 114,14,120,48, ListBox1$, .ListBox1
        ComboBox 113,68,120,84, ComboBox1$, .ComboBox1
    End Dialog

    'Prepare to display dialog box.
    '-----
    'Declare the dialog type using Dim ... as UserDialog command
    Dim aSampleDialog as UserDialog

    'Load the list box and combo box arrays with app window names
    AppList ComboBox1$
    AppList ListBox1$

    'Load the text box with an initial value
    aSampleDialog.TextBox1 = "123456789012345678901234567890"
    'Note that you reference a dialog box field as follows:
    '    <DialogBoxName>.<FieldName>

    'Display the Dialog
    '-----
    a% = Dialog(aSampleDialog)
        'Returns integer indicating button chosen

```

```

'Interpret user input
'-----
'This example builds a string describing the contents of the
'dialog box when the user finished making changes.
crlf$ = chr$(13)+chr$(10)
dlgstr$ = "Button pushed = "
'Determine button pushed.
select case a%
    case -1
        dlgstr$ = dlgstr$ + "OK"
    case 0
        dlgstr$ = dlgstr$ + "Cancel"
    case 1
        dlgstr$ = dlgstr$ + "PushButton1"
    case 2
        dlgstr$ = dlgstr$ + "PushButton2"
end select
dlgstr$ = dlgstr$ + crlf$
'Determine new value of each dialog box component
dlgstr$ = dlgstr$ + "Option = " + ↵
    str$(aSampleDialog.OptionGroup1) + crlf$
dlgstr$ = dlgstr$ + "Checkbox = " + ↵
    str$(aSampleDialog.CheckBox1) + crlf$
dlgstr$ = dlgstr$ + "TextBox = " + aSampleDialog.TextBox1 + crlf$
dlgstr$ = dlgstr$ + "ListBox = " + ↵
    ListBox1$(aSampleDialog.ListBox1) + crlf$
dlgstr$ = dlgstr$ + "ComboBox = " + aSampleDialog.ComboBox1
'Display the dlgstr$ string in a message box.
msgbox dlgstr$
end sub

```

Exiting the Dialog Editor

When you have finished creating or modifying the dialog box template, choose Exit & Update from the File menu. If you created a new dialog template, the dialog template commands will be added to your script. If you modified an existing dialog template from your script, the old template is replaced with the new one.

If you do **not** want to update your script, choose Exit from the File menu. You will be asked whether to update your script before the connection to the **Dialog Editor** is terminated. Answer No.

Shortcut Keys

You can use the following keys when working with dialog box controls:

Key	Description
Ctrl+C	Copy the current control to the clipboard
Ctrl+V	Paste the contents of the clipboard
Ctrl+X	Cut the current control and store it in the clipboard
Ctrl+D	Duplicate the current control
Del	Clear the current control
Ctrl+I	Display the Information dialog box
F1	Go to the Dialog Editor Help Contents
Shift+F1	Context-sensitive Help for menus and Toolbar buttons
F2	Toggle dialog box test on and off

