

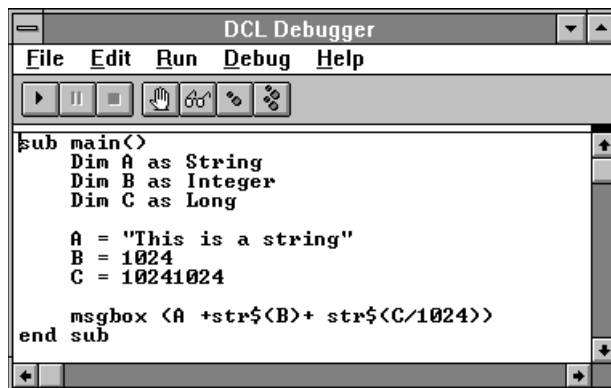
Using the Debugger

After you have checked the syntax of your script, the **Debugger** can help you find logic errors. The **Debugger** lets you set breakpoints, establish watch variables, and trace through scripts statement by statement.

Starting the Debugger

Before starting the **Debugger**, use the **DCL Editor Tools/Syntax Check** command to ensure that your script is free of syntax errors. The **Debugger** will not run or trace through a script with syntax errors.

To start the **Debugger**, choose the **DCL Editor Tools/Debugger** command.



The Debugger window

Running a Script

You can run your script at full speed to reach the point where you want to begin debugging.

Before running the script, set a breakpoint (as described under “Setting Breakpoints” in this chapter) where you want to begin debugging. You can set breakpoints throughout your script.

To run a script at full speed, choose the Start command from the Run menu. The script runs until it encounters the first breakpoint. (You must use the Run/Arguments command in the **DCL Editor** window to set command-line arguments, which are passed to `sub main`.)

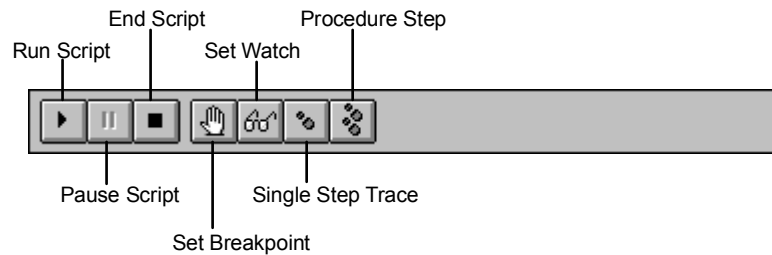
At this point you can trace through the script as described under “Tracing Through a Script” in this chapter or continue at full speed to the next breakpoint by choosing the Continue command from the Run menu.

To pause the execution of the script, choose the Break command from the Run menu.

To stop the script, choose the End command from the Run menu.

Using the Toolbar

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



The Debugger Toolbar

Tracing Through a Script

Tracing means executing your script one line at a time. You can trace through your script from the beginning or run the script at full speed to the point where you want to begin debugging (as described under “Running a Script” in this chapter). While you are tracing through a script, the current line (called the *instruction pointer*) is highlighted.

To execute the current line and move the instruction pointer to the next line, choose the Single Step command from the Debug menu.

If you want to skip over a user-defined function or subroutine, choose the Procedure Step command from the Debug menu. This command is the same as the Single Step command, except that it does not trace into user-defined subroutines or functions.

To reposition the instruction pointer, move the cursor to the line you want to make current and choose the Set Next Statement command from the Debug menu.

Setting Breakpoints

To set a breakpoint, position the insertion point on the line where you want to set the breakpoint and choose Toggle Breakpoint from the Debug menu. Do not set a breakpoint on a line that contains no code.

To toggle a breakpoint off, position the insertion point on the line where the breakpoint has been set and choose Toggle Breakpoint from the Debug menu.

To remove all breakpoints, choose Clear All Breakpoints from the Debug menu.

You can set up to 255 breakpoints in a script.

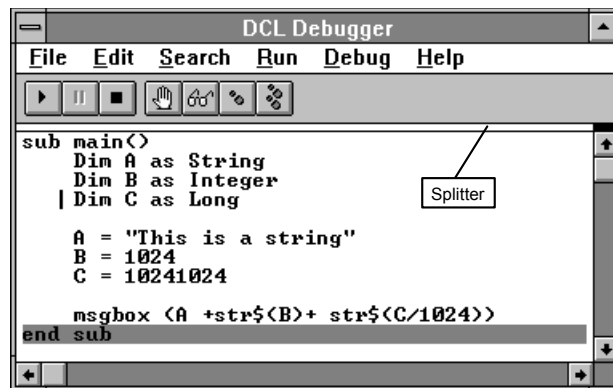
All breakpoints are removed when you exit the **Debugger**.

Using Watch Variables

Watch variables appear in the watch pane of the **Debugger** window. As you step through your script, the current value of the variable and other information are displayed in the watch pane. If the variable is not currently in scope, the message <Not in Context> appears.

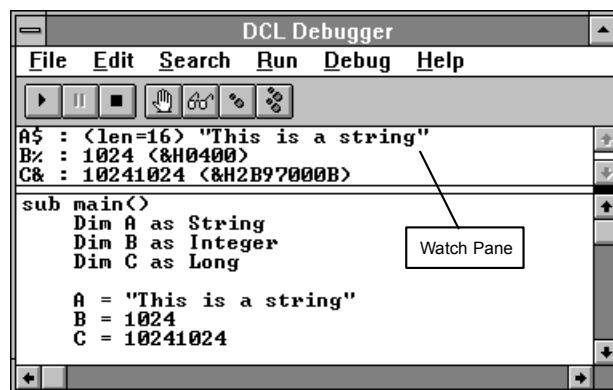
Opening the Watch Pane

When you start the **DCL Debugger** the watch pane is not open. The *splitter* (pictured in the following window) for the closed pane appears just below the Toolbar.



Window with the watch pane closed

To open the *watch pane* (pictured in the following window), click inside the splitter, hold the mouse button down and drag the pane to the desired size.



Window with the watch pane open

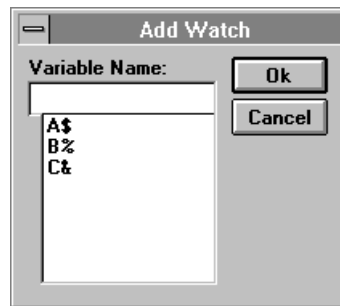
Adding a Watch Variable

For a variable to appear in the watch pane, you must explicitly add it.



To add a watch variable

1. Choose the Add Watch command from the Debug menu.



2. Enter the name of the variable you want to watch or select it from the list box. (The variable will not show up in the Add Watch dialog box until you have begun to step through the script.)
3. Choose OK.

Once added the variable appears in the watch pane.

Deleting a Watch Variable

You can delete variables from the Watch Pane.



To delete a variable from the watch pane

1. Select the variable in the watch pane.
2. Choose the Delete Watch command from the Debug menu or press the DEL key.

Editing Text

The **Debugger** provides the basic text editing functions.

Use the Cut command on the Edit menu to remove a block of text you have selected and store it in the Windows clipboard.

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

Use the Paste command to insert the text currently in the clipboard at the insertion point.

Use the Clear command to remove the block of text you have selected. It is not written to the clipboard. You cannot undo the Clear command.



To select a block of text for an editing operation

1. Position the insertion point at the beginning of the text to be selected and click the left mouse button.
2. Move the insertion point to the end of the block to be selected.
3. Hold down the Shift key and press the left mouse button.

OR

1. Move the insertion point to the beginning of the text to be selected.
2. Hold the Shift key.
3. Move the cursor to the end of the text to be selected.

Finding Text

Use the Find command to search for a particular text string in the script.



To find a string of text in a script

1. Choose the Find command from the Search menu.



2. Enter the text string to search for in the Find What text box.
3. Select the Match Case check box if you want the search to differentiate between upper- and lowercase letters.
4. Specify the Direction of the search—up or down.
5. Choose the Find Next button to begin the search.

The first occurrence of the string after the insertion point is highlighted.
6. To continue searching for subsequent occurrences of the string, choose Find Next.

The next occurrence of the string is highlighted.

Note: You can also continue the search by closing the dialog box (by choosing Cancel) and choosing Find Next from the Search menu.

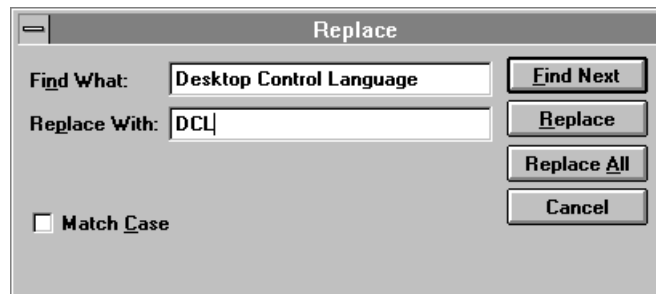
Replacing Text

Use the Replace command to replace one text string with another in the current script.



To replace one text string with another

1. Choose the Replace command from the Search menu.



2. Enter the text string to search for in the Find What text box.
3. Enter the replacement text in the Replace With box.
4. Select the Match Case check box if you want the search-and-replace operation to differentiate between upper- and lowercase letters.
5. Choose the Find Next button to begin the search.
The first occurrence of the string after the insertion point is highlighted.
6. If you want to substitute the replacement text for the string, choose the Replace button. (The search for the next matching string continues after the replacement is made.)

OR

If you want to go on to the next occurrence of the string without making the text change, choose the Find Next button.

OR

If you want to change all occurrences of the string to the replacement text, choose the Replace All button. (To avoid accidental changes, we recommend that you use this feature with caution.)

Editing Dialogs

You can use the **Dialog Editor** (described in Chapter 4) to insert a new dialog box template in the script you are debugging or update an existing dialog box template from your script.



To create a new dialog box template to insert in your script

1. Choose the Insert New Dialog command from the Edit menu.
2. Use the **Dialog Editor** to create the dialog box template.
3. Use the **Dialog Editor** Exit & Update command to exit the **Dialog Editor** and insert the template in the script.



To update a dialog box template from your script

1. Select the text containing the dialog box template.
2. Choose the Edit Dialog command from the Edit menu.
3. Use the **Dialog Editor** to modify the dialog box template.
4. Use the **Dialog Editor** Exit & Update command to exit the **Dialog Editor** and replace the old template in the script.

Getting Help

DCL provides extensive Help that describes how to use the **Debugger**.

To display the Table of Contents for the **Debugger** Help, choose Contents from the Help menu.

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, **DCL** displays a particular Help topic, based on your current activity.

For Help on a particular **DCL** language element (statement, function, or system variable) enter the language element in the document and press F1.

For help on a menu item or Toolbar button:

1. Press Shift + F1.

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

2. Choose the menu and command, or Toolbar button you want Help on.

Exiting the Debugger

When you have finished debugging your script, choose Exit & Update from the File menu. Your script will be updated with any changes you made during the debugging process.

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 **Debugger** is closed. Answer No.

Shortcut Keys

You can use the following keys when debugging scripts:

Key	Description
ESC	Closes the Debugger
CTRL+C or CTRL+Insert	Copy
Ctrl+V or SHIFT+Insert	Paste
Ctrl+X or SHIFT+Delete	Cut
DEL	Clear
	OR
	Deletes the current Watch variable if the Watch window is active
F3	Find Next
SHIFT+F8	Procedure Step
F8	Single Step
F9	Toggle Breakpoint on or off

Continued...

Key	Description
F6	Moves between the Watch window and the editing workspace if the Watch window is open
F1	Context-sensitive Help for current command in editing window
SHIFT+F1	Context-sensitive Help for menus and Toolbar buttons