

CHAPTER 19

Debugging JavaScript Code

Macromedia Dreamweaver JavaScript Debugger allows you to uncover errors in your client-side JavaScript code. You can write the code using Dreamweaver Code view (or Code inspector), then run the debugger to check your code for syntax and logical errors. A syntax error causes the browser to report an error message; a logical error causes your page to function incorrectly, but is not reported by the browser. The debugger works with Microsoft Internet Explorer and Netscape Navigator on the Windows platform and Netscape Navigator on the Macintosh platform. For more information about writing scripts, see “Inserting scripts” on page 335.

The debugger checks your code for syntax errors first, then runs with the browser to help you check for logical errors. If you have logical errors, the JavaScript Debugger window helps you isolate the errors in your JavaScript code by letting you examine variables and document properties while your program is running. You can set breakpoints (similar to alert statements) in your code to stop the execution of the program and display the values of JavaScript objects and properties in a variable list. You can also step to the next statement or step into a function call to see the variable values change.

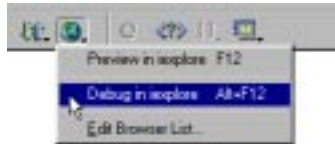
The JavaScript Debugger can significantly reduce the time it takes you to find and isolate errors in your code.

Running the debugger

After you write your code, you can start the JavaScript Debugger to check for errors. The debugger checks for syntax errors first, then opens your page in the browser so you can check for logical errors.

To start debugging:

- 1 Choose File > Debug in Browser, then select the browser from the list. Alternatively, click the Preview/Debug in Browser button in the toolbar (View > Toolbar) and select Debug in Internet Explorer or Debug in Netscape Communicator.



If the debugger finds syntax errors, it stops and lists them in the JavaScript Syntax Errors window. See “Finding syntax errors” on page 475.

- 2 If you are using Netscape Navigator, click OK in the debugger warning box that appears, then click Grant in the Java Security dialog box.

Note: If you have already accepted a Macromedia Security Certificate, the Java Security dialog box may not appear.

- 3 If you are using Internet Explorer (Windows only), click Yes in the Java Security dialog box, then OK in the debugger warning box that appears.

The debugger connects with the browser, but does not actually make a network connection or connect to any Internet servers. The browser appears with the JavaScript Debugger window, which is stopped automatically at the first line of code.

The JavaScript Debugger window appears with the browser window. The debugger stops automatically at the first line of code.

To run the debugger:

Click the Run button in the JavaScript Debugger window.

To stop the debugger:

Click the Stop Debugging button in the JavaScript Debugger window. The debugger will close.

Finding syntax errors

If the debugger finds syntax errors, it stops and lists the errors in the JavaScript Syntax Errors window.

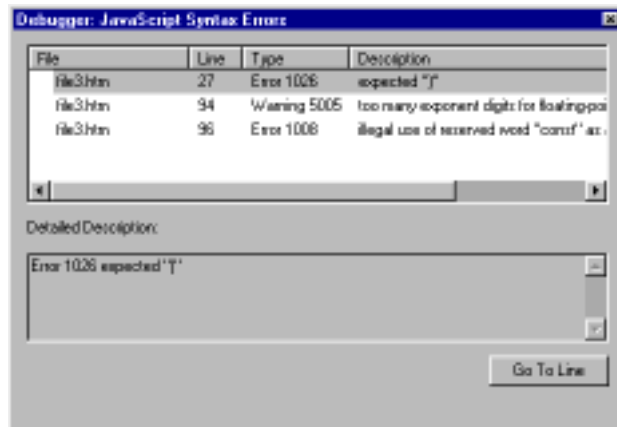
To view the error description:

Select an error in the JavaScript Syntax Errors window. A description of the error will appear in the Detailed Description area.

To go to the selected error in your code, do one of the following:

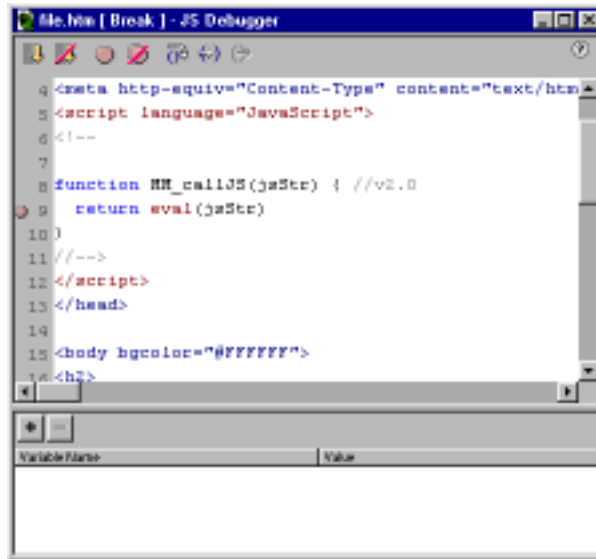
- Double-click the error.
- Click the Go To Line button.

The code is highlighted in the Code view (or Code inspector).

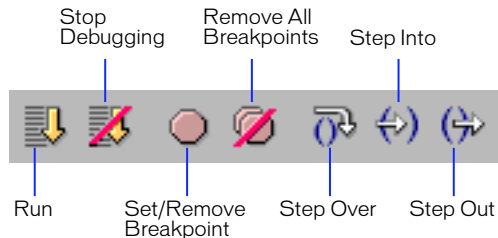


Finding and fixing logical errors

When the debugger finds logical errors, the JavaScript Debugger window opens. A breakpoint (similar to an alert) is automatically set in the first line of your code. The debugger stops executing at each breakpoint, giving you the opportunity to view the values of JavaScript objects and properties in the variable list window.



After the debugger stops at a breakpoint, you can step through your code (execute one statement at a time). This allows you to see if the program executes as it should. The debugger can even step into linked code. For example, if your code contains a link to a source file, the debugger steps into the source file and displays it in the JavaScript Debugger window. As you step through the code, you can watch as the values of your variables change through your program.



Setting breakpoints

A breakpoint marks a spot in the code where you want the program execution to stop. When you set a breakpoint, it is marked with a small red dot in the left margin of the JavaScript Debugger window. When the program stops executing at that breakpoint, a small arrow appears over the dot, and you can examine the objects and properties that exist at that point. This allows you to quickly pinpoint the source of the bug in the JavaScript code.

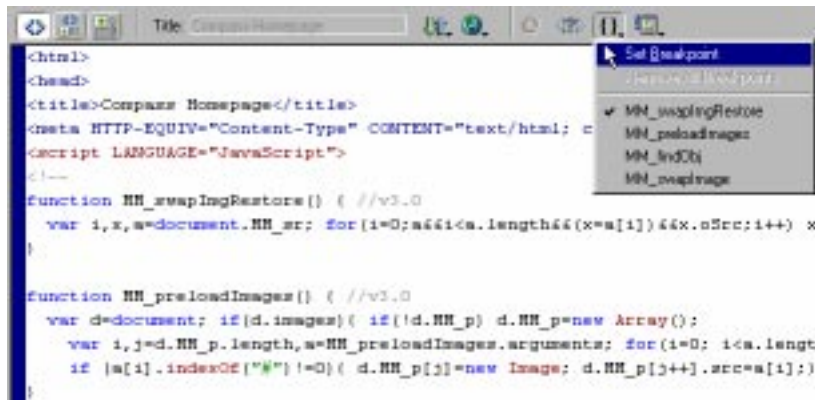
Breakpoints can only be set in the JavaScript code (between `script` tags) or on a line with an event handler. If you set a breakpoint elsewhere, Dreamweaver will automatically set the breakpoint on the next valid line of code (or place the insertion point there, if the line already has a breakpoint set on it). If there are no valid lines on which a breakpoint can be set, you will hear a beep.

To set a breakpoint, do one of the following:

- In the JavaScript Debugger window, place the insertion point in the line where you want the breakpoint, then click the Set/Remove Breakpoint button at the top of the debugger window. To remove the breakpoint, click the Set/Remove Breakpoint button again.



- In the Code view (or Code inspector), place the insertion point in the line where you want the breakpoint, then select Edit > Set Breakpoint or choose Set Breakpoint from the Code Navigation pop-up menu in the toolbar. You can also right-click (Windows) or Control-click (Macintosh) and choose Set Breakpoint from the context menu. To remove the breakpoint, select Remove Breakpoint from the context menu.



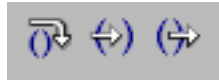
To remove all breakpoints, do one of the following:

- In the JavaScript Debugger window, click the Remove All Breakpoints button.
- In the Code view (or Code inspector), select Remove All Breakpoints from the Code Navigation pop-up menu in the toolbar or choose Edit > Remove All Breakpoints.

Stepping through code

You can step through your code to execute your statements one at a time and monitor their effects in your program. For example, you can step over an `if` condition and see if the program stops at the first line within the conditional statement or at the next executable line after the `if` statement.

When the debugger stops at a statement with a function call, you can check the function to make sure it executes correctly. If the function is correct, you can step out of it to allow the debugger to run until the function returns. The program will stop again at the next statement after the function call location. If you attempt to step into a statement that contains a nonstandard JavaScript function, the behavior will be that of stepping over.



To step over a statement:

Click the Step Over button at the top of the JavaScript Debugger window.

When the program stops at any statement (including those with a function call), you can step over that statement to continue and pause before the next statement.

To step into a function:

Click the Step In button at the top of the JavaScript Debugger window.

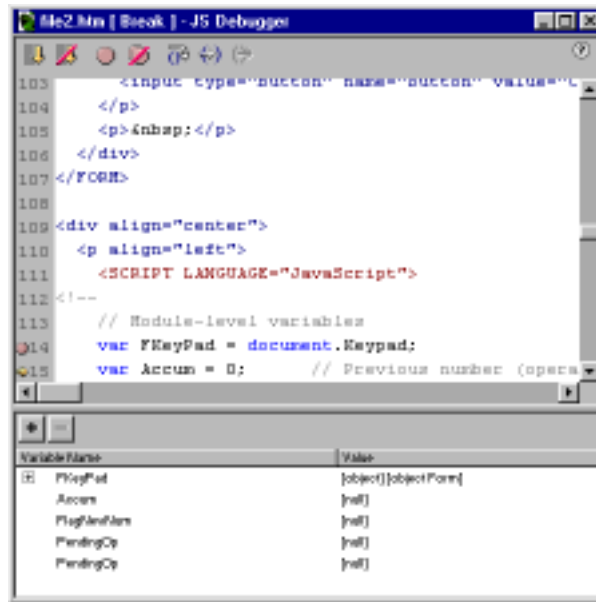
To step out of a function:

Click the Step Out button.

You can only use Step Out when the debugger is within a user-defined function. Stepping out causes the remaining statements in the function definition to be executed. The debugger will pause at the next statement.

Watching and editing variable values

To check the values of your variables as you step through the code, you use the Dreamweaver variable list located in the bottom pane of the JavaScript Debugger window. You enter variable names in the left column; the right column lists the current values of each variable when the program stops execution at a breakpoint or after you've stepped in the code.



To add a variable to your variable list, do one of the following:

- Select the name of the variable in the code portion of the JavaScript Debugger window. Click the Plus (+) button and press Enter.
- Click the Plus (+) button, type the name of the variable you want to watch, and press Enter.

The values will appear next to each variable as you step through the code. If the variable is an object with properties, you can expand the variable (show its properties and values) by clicking the Plus (+) button (Windows) or triangle button (Macintosh) next to it in the list. The expanded variable is automatically collapsed each time you step through the code.

To remove a variable list item:

- 1 Select the item in the variable list.
- 2 Click the Minus (-) button.

To edit a value:

- 1 Select the item in the variable list.
- 2 Click the value in the value list.
- 3 Edit the value by typing in the text box that appears.

