example would be to use the Continue Regardless of Result setting and execute some sort of logging functionality each time the breakpoint is encountered. You could even use Ajax techniques to send the logging information to the server!

Local Variables List

The Local Variables window allows you to inspect and even modify variable values during script execution. The Local Variables window always displays all the variables within scope whenever a breakpoint is encountered and execution of the script is suspended.

The Local Variables window always has two top-level items, Scope and This. Scope refers to all the variables within the nearest current scope of execution. Because most JavaScript is written as a function, the nearest scope is usually `function` scope. For example, if a breakpoint within a function is encountered, then the Scope item within the Local Variables window will refer to all variables that are within that function's scope—namely, any variable defined with the keyword `var` within that function. Variables defined in the global scope (those defined outside any function body) are technically accessible within functions, but they are not shown within the current variable scope (see Figure 7-22).

```
function createQueryString() {
  var searchString = document;
  searchString = escape(searchString);
  var maxResultsCount = document;
  var queryString = "query=" + maxResultsCount;
  return queryString;
}

function handleStateChange() {
  if (XmlHttp.readyState == 4)
```

![Figure 7-22. All the variables within the current scope](image)

The second top-level item displayed in the Local Variables window is the `this` item. The `this` item refers to whichever object the keyword `this` refers. If the breakpoint occurs within a function that is part of an object, then `this` refers to the current object instance. The normal reference for `this` is the browser's `window` object. Note that any variables defined within the global scope will appear under the `this` item.

The Local Variables window uses small icons next to a variable's name to indicate the variable's data type. The data types available are boolean, double, integer, null, object, string, and void. Figure 7-23 shows the Local Variables window with each data type. Note the icon that is used for each data type.