Other than the variable named `objVar` in Figure 7-23, all the data types are simple, nonobject data types, and their values are easily discernable in the Value column of the Local Variables window. Note how the `objVar` variable has a plus sign next to it, indicating that it can be opened to inspect the object’s properties. Figure 7-24 shows the Local Variables window using an object variable type. Note that in this instance the breakpoint is set within an object’s member function, meaning that within the Local Variables window the `this` item refers to the object instance itself while the scope item has no child items.

In this instance, the object has two properties, `curbWeightInPounds` and `wheelCount`, and their respective values are displayed within the Local Variables window.

Up until now you’ve probably been using alert boxes to show the values of variables during runtime. While this technique certainly works, it takes time to pepper your script with alerts, and each alert usually shows only one variable value at a time. After the script is debugged, you’ll need to remove all those alert boxes, and if you ever need to debug again, you’ll end up adding alert boxes again. With Venkman, all you have to do is set a breakpoint and use the Local Variables window to inspect the values of variables, and they are all there for quick access.

Step back for a moment to think about how the Local Variables window makes Venkman an extremely powerful debugging tool. With one quick glance you are able to view the values of all the objects and variables that the JavaScript engine knows about. Using breakpoints, you