Go

Demos
Technical Papers
Tips and Techniques
Videos

Email Newsletter
Developer Team Blog
Facebook
Twitter

Links to Related Sites
Microsoft Access Developer Help Center
Microsoft Access Query Help Center
Book Recommendations
My FMS

All Our Microsoft Access Products

Thank you! Thank you! I just finished reading this document, which was part of a link in the recent Buzz newsletter. I have printed it for others to read, especially those skeptical on the powers of Access and its capabilities.
Darren D.

Build an Evaluate Function

Provided by: Dan Haught, Executive Vice President

One of the more useful features in prior versions of Visual Basic was the Eval() function. Eval would evaluate the supplied string and return a result. This resulted in the ability to evaluate arbitrary statements at runtime. Unfortunately, neither C# nor Visual Basic .NET has the same functionality.

Fortunately, the power of .NET, specifically with the ability to call components authored in other .NET languages makes this fairly simple to implement. Our first piece of the puzzle is JScript.NET, the .NET version of JavaScript. JScript.NET does include an Eval function. The question is how to get to that function from our C# or Visual Basic .NET projects.

To build your own Eval equivalent, follow these steps.

Step 1: Create a JScript Wrapper DLL

1. Open Notepad and type in the following code:

```
class JScriptEval
 {
   function Evaluate(evalString : String)
   {
     return eval(evalString);
   }
 }
```

2. Save the file as jScriptEval.js and close NotePad.

3. Compile the file using the .NET JavaScript compiler. The following command line shows how to do this. Note that your path will need to be the directory where the compiler lives. Typically, this is something like:

```
C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705

jsc /t:library jScriptEval.js
```

4. If everything worked according to plan, your directory should now contain a file named jScriptEval.dll.

Step 2: Create References in your Visual Studio .NET project

1. Back in your Visual Basic.NET or C# project, create a reference to your new DLL. To do this, right-click on your project, in the Solution Explorer, Select the .NET tab, press the Browse button to locate JScriptEval.DLL and press [OK]

2. You will also need a reference to the Microsoft JScript library. This is typically located in c:\windows\microsoft.net\framework\v1.0.3705\Microsoft.JScipt.dll.

Step 3: Test your new eval code

Test your eval library like this:

```
Dim j As New JScriptEval
Dim s1 As String = "1 + 2 * 3"
MsgBox(j.Evaluate(s1))
```

Security Issues

You should be aware that using Eval statements can lead to potential security risks. Eval can execute just about any program code. There are two ways to

minimize this. First, you can use a regular expression evaluator to make sure that the string passed to Eval contains only letters, operators and numbers, not words. Secondly, consider using .NET security features to disallow any potentially dangerous operations before code is executed by Eval.

Return to the tips page