



Faster DHTML in 12 Steps

The introduction of Dynamic HTML (DHTML) in Microsoft Internet Explorer 4.0 made available a new programming model to Web authors and developers. Since then, Web authors have taken advantage of this powerful feature to provide dynamic content, styles, and positioning, enabling a rich interactive experience for the Web user. Because of the flexibility of DHTML, there is often more than one way to accomplish what you want to do. Understanding how the HTML parsing and rendering component of Windows Internet Explorer processes your requests can give you the edge when deciding which methods work best for the job. This article describes how using some DHTML features can affect performance more than others, and it presents tips that will help your pages perform faster.

- [Batch Your DHTML Changes](#)
- [Talk to Your innerText](#)
- [Use the DOM to Add Individual Elements](#)
- [Expand Your Options in a SELECT Element](#)
- [Use the DOM to Update Tables](#)
- [Write Once, Use Many Times](#)
- [Don't Be Too Dynamic with Your Properties](#)
- [Data Binding is Good for You](#)
- [Keep Your Expando Properties Off the Document](#)
- [Avoid switching classes and style rules](#)
- [Collapse a Text Range Before Finding the Parent](#)
- [Related Topics](#)

Batch Your DHTML Changes

The most significant way of increasing performance on a DHTML Web page is to improve how you apply changes to the HTML content on your page. It is important to understand that there are many ways to update a Web page. From client scripting, the Web author can apply blocks of HTML text, or access individual HTML elements, using the [DHTML object model](#) [[http://msdn.microsoft.com/en-us/library/ms533022\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533022(VS.85).aspx)] or the [World Wide Web Consortium \(W3C\) Document Object Model \(DOM\)](#) [[http://msdn.microsoft.com/en-us/library/ms533043\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533043(VS.85).aspx)]. Whenever you make a change to the HTML content, the HTML parsing and rendering component of Internet Explorer must reorganize its internal representation of the page, recalculate the layout and flow of the document, and display the changes. Although actual performance depends on the changes you make and the content of the Web page, these operations are among the more expensive. If you are applying a block of HTML text, as opposed to accessing individual elements, then the HTML parser must be invoked, which will incur an additional performance cost. Methods and properties that accept HTML text include the [insertAdjacentHTML](#) [[http://msdn.microsoft.com/en-us/library/ms536452\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536452(VS.85).aspx)] and [pasteHTML](#) [[http://msdn.microsoft.com/en-us/library/ms536656\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536656(VS.85).aspx)] methods, and the [innerHTML](#) [[http://msdn.microsoft.com/en-us/library/ms533897\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533897(VS.85).aspx)] and [outerHTML](#) [[http://msdn.microsoft.com/en-us/library/ms534310\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms534310(VS.85).aspx)] properties.

Tip 1: Make changes to the HTML content in one script function. If your design makes use of multiple event handlers, such as responding to mouse movements, make your changes in one place.

Another important fact about the HTML parsing and rendering component: it recalculates layout and displays the Web page when any script returns control (for example, when a script event handler function exits, or when a method such as [setTimeout](#) [[http://msdn.microsoft.com/en-us/library/ms536753\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536753(VS.85).aspx)] is called). Now that you understand something about how Internet Explorer processes changes, you can begin improving the performance of your Web page.

Tip 2: Build a string of HTML and make one change to the document, rather than making multiple updates. If HTML content is not required, consider using the [innerText](#) [[http://msdn.microsoft.com/en-us/library/ms533899\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533899(VS.85).aspx)] property.

In the following examples, the slow method invokes the HTML parser each time the **innerHTML** property is set. To improve performance, a string can be built, which is then assigned to the **innerHTML** property.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip2.htm>
[<http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip2.htm>]

Slow:

[Copy Code](#)

```
divUpdate.innerHTML = "";
for ( var i=0; i<100; i++ )
{
    divUpdate.innerHTML += "<SPAN>This is a slower method! </SPAN>";
}
```

Fast:

[Copy Code](#)

```
var str="";
for ( var i=0; i<100; i++ )
{
    str += "<SPAN>This is faster because it uses a string! </SPAN>";
}
divUpdate.innerHTML = str;
```

For more information on dynamic content, please see [About Dynamic Content](#) [[http://msdn.microsoft.com/en-us/library/ms533040\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533040(VS.85).aspx)] .

Talk to Your innerText

In the DHTML object model, text content for an HTML element is accessed through the **innerText** property, whereas the W3C DOM provides a separate child text node. It is faster to update the content of an element directly through the **DHTMLinnerText** property than it is to call the DOM [createTextNode](#) [[http://msdn.microsoft.com/en-us/library/ms536400\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536400(VS.85).aspx)] method.

Tip 3: Use the **innerText** property to update text content.

The following examples show how to improve performance through the use of the **innerText** property.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip3.htm>
[<http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip3.htm>]

Slow:

[Copy Code](#)

```
var node;
for (var i=0; i<100; i++)
{
    node = document.createElement( "SPAN" );
    node.appendChild( document.createTextNode( " Using createTextNode() " ) );
    divUpdate.appendChild( node );
}
```

Fast:

[Copy Code](#)

```
var node;
for (var i=0; i<100; i++)
{
    node = document.createElement( "SPAN" );
    node.innerText = " Using innerText property ";
    divUpdate.appendChild( node );
}
```

Use the DOM to Add Individual Elements

As mentioned previously, accessing methods that apply HTML text causes the HTML parser to be invoked, with subsequent performance loss. It follows that it is faster to add elements using the [createElement](http://msdn.microsoft.com/en-us/library/ms536389(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536389\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536389(VS.85).aspx)] and [insertAdjacentElement](http://msdn.microsoft.com/en-us/library/ms536451(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536451\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536451(VS.85).aspx)] methods than to make a single call to the **insertAdjacentHTML** method.

Tip 4: Calling the **createElement** and **insertAdjacentElement** methods can often be faster than calling the **insertAdjacentHTML** method.

Batching the DHTML updates and performing a single call to the **insertAdjacentHTML** method would improve performance, but there might be occasions when creating elements directly through the DOM is more efficient. This is one scenario where you might try both methods and determine which is faster.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip4.htm>
[<http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip4.htm>]

Slow:

```
for (var i=0; i<100; i++)
{
    divUpdate.insertAdjacentHTML( "beforeEnd", "<SPAN> Uses insertAdjacentHTML()
</SPAN>" );
}
```

[Copy Code](#)

Fast:

```
var node;
for (var i=0; i<100; i++)
{
    node = document.createElement( "SPAN" );
    node.innerHTML = " Uses insertAdjacentElement() ";
    divUpdate.insertAdjacentElement( "beforeEnd", node );
}
```

[Copy Code](#)

Expand Your Options in a **SELECT** Element

An exception to our earlier rule about using the HTML text methods is when you add a large number of [OPTION](http://msdn.microsoft.com/en-us/library/ms535877(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535877\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535877(VS.85).aspx)] elements to a [SELECT](http://msdn.microsoft.com/en-us/library/ms535893(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535893\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535893(VS.85).aspx)] element. It is more efficient to use the **innerHTML** property than to call the **createElement** method to access the options collection.

Tip 5: Use **innerHTML** to add a large number of options to a **SELECT** element.

Use string concatenation to build up the **SELECT** element HTML text, then use this to set the **innerHTML** property. For very large numbers of options, string concatenation can also affect performance. In this situation, build an array and call the Microsoft JScript join method to perform a final concatenation of the **OPTION** element HTML text.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip5.htm>
[<http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip5.htm>]

Slow:

```
var opt;
divUpdate.innerHTML = "<SELECT ID='selUpdate'></SELECT>";
for (var i=0; i<1000; i++)
```

[Copy Code](#)

```
{
  opt = document.createElement( "OPTION" );
  selUpdate.options.add( opt );
  opt.innerText = "Item " + i;
}
```

Fast:

[Copy Code](#)

```
var str("<SELECT ID='selUpdate'>";
for (var i=0; i<1000; i++)
{
  str += "<OPTION>Item " + i + "</OPTION>";
}
str += "</SELECT>";
divUpdate.innerHTML = str;
```

Faster:

[Copy Code](#)

```
var arr = new Array(1000);
for (var i=0; i<1000; i++)
{
  arr[i] = "<OPTION>Item " + i + "</OPTION>";
}
divUpdate.innerHTML = "<SELECT ID='selUpdate'>" + arr.join() + "</SELECT>";
```

Use the DOM to Update Tables

It is more efficient to insert table rows and cells using DOM methods than to use the [insertRow](http://msdn.microsoft.com/en-us/library/ms536457(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536457\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536457(VS.85).aspx)] and [insertCell](http://msdn.microsoft.com/en-us/library/ms536455(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536455\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536455(VS.85).aspx)] methods that are part of the DHTML table object model. This is especially true when creating large tables.

Tip 6: Use DOM methods to build large tables.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip6.htm>
[<http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip6.htm>]

Slow:

[Copy Code](#)

```
var row;
var cell;
for (var i=0; i<100; i++)
{
  row = tblUpdate.insertRow();
  for (var j=0; j<10; j++)
  {
    cell = row.insertCell();
    cell.innerText = "Row " + i + ", Cell " + j;
  }
}
```

Fast:

[Copy Code](#)

```
var row;
var cell;
var tbody = tblUpdate.childNodes[0];
tblUpdate.appendChild( tbody );
for (var i=0; i<100; i++)
{
    row = document.createElement( "TR" );
    tbody.appendChild( row );
    for (var j=0; j<10; j++)
    {
        cell = document.createElement( "TD" );
        row.appendChild( cell );
        cell.innerText = "Row " + i + ", Cell " + j;
    }
}
```

Write Once, Use Many Times

If your Web site uses script to perform common operations, consider placing these functions in a separate file that can be reused by more than one Web page. As well as improving code maintenance, the script file will remain in the browser cache and will need to be downloaded to the user's machine only once during their visit to your site. You can obtain the same benefits by placing your common style rules in a separate file.

Tip 7: Reuse your script by placing common code in a behavior or a separate file.

For even better reuse of script functions, place your common script operations in a DHTML attached or element [behavior](http://msdn.microsoft.com/en-us/library/ms531079(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms531079\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms531079(VS.85).aspx)] . Behaviors provide a great way to reuse script and build components that can be accessed from HTML, enabling you to extend the DHTML object model with your own objects, methods, properties and events. For behaviors that do not make use of the [viewlink](http://msdn.microsoft.com/en-us/library/ms531428(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms531428\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms531428(VS.85).aspx)] feature, consider using the [lightweight behavior](http://msdn.microsoft.com/en-us/library/ms531426(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms531426\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms531426(VS.85).aspx)] feature available in Internet Explorer 5.5 for even more efficient code encapsulation. In addition, you will see better performance if your script code is in a single [SCRIPT](http://msdn.microsoft.com/en-us/library/ms535892(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535892\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535892(VS.85).aspx)] block.

Don't Be Too Dynamic with Your Properties

[Dynamic properties](http://msdn.microsoft.com/en-us/library/ms537634(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms537634\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms537634(VS.85).aspx)] provide a way for Web authors to use an expression as a property value. This expression is evaluated at run time, and the resulting value is applied to the property. This is a powerful feature. It can be used to reduce the amount of script on the page, but it can also have an adverse effect on performance because the expressions must be recalculated regularly and often have dependencies on other property values. This is especially true with positioning properties.

Tip 8: Restrict your usage of dynamic properties.

Data Binding is Good for You

[Data Binding](http://msdn.microsoft.com/en-us/library/ms531387(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms531387\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms531387(VS.85).aspx)] is a powerful feature that enables you to bind the results of a database query, or the contents of an XML data island, to HTML elements on your Web page. You can provide data sorting and filtering functionality, and different views of the data, without returning to the server to fetch the data. Imagine a Web page that presents company data as a line, bar or pie chart, in addition to buttons to sort the data by office, product or sales period. All this functionality can be provided with only one visit to the server—now that's great performance.

Tip 9: Use data binding to provide rich client-side data views.

For more information on data binding, please see the following articles:

- [Introduction to Data Binding](http://msdn.microsoft.com/en-us/library/ms531387(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms531387\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms531387(VS.85).aspx)]
- [Getting Your Page's Data into a Bind](http://msdn.microsoft.com/en-us/library/Bb264002(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/Bb264002\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/Bb264002(VS.85).aspx)]
- [Lean, Mean, and Virtually Data Bound](http://msdn.microsoft.com/en-us/library/Bb263972(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/Bb263972\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/Bb263972(VS.85).aspx)]

Keep Your Expando Properties Off the Document

[Expando](http://msdn.microsoft.com/en-us/library/ms533747(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms533747\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533747(VS.85).aspx)] properties are arbitrary properties that can be added to any object. They are useful for storing information within the current Web page and are another way to extend the DHTML object model. For example, you could assign a clicked property on your HTML elements and use this to indicate to the user which elements have been clicked. You can also use **expando** properties when firing events, to provide additional context information to the event handler function. Whatever use you have for **expando** properties, be sure not to set them on the [document](http://msdn.microsoft.com/en-us/library/ms531073(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms531073\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms531073(VS.85).aspx)] object. If you do, the document must perform additional recalculations when you access the property.

Tip 10: Set **expando** properties on the [window](http://msdn.microsoft.com/en-us/library/ms535873(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535873\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535873(VS.85).aspx)] object.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip10.htm>
[<http://samples.msdn.microsoft.com/workshop/samples/author/perf/tip10.htm>]

Slow:

```
for (var i=0; i<1000; i++)
{
    var tmp;
    window.document.myProperty = "Item "+i;
    tmp = window.document.myProperty;
}
```

[Copy Code](#)

Fast:

```
for (var i=0; i<1000; i++)
{
    var tmp;
    window.myProperty = "Item "+i;
    tmp = window.myProperty;
}
```

[Copy Code](#)

Avoid switching classes and style rules

Switching classes and style rules can be an expensive operation, requiring recalculation and layout of the entire document. If your Web site uses style sheets to provide alternative views of your content, consider modifying the [style](http://msdn.microsoft.com/en-us/library/ms536498(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536498\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536498(VS.85).aspx)] object directly for the elements you wish to change, rather than modifying an element's [className](http://msdn.microsoft.com/en-us/library/ms533560(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms533560\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533560(VS.85).aspx)] property or the [styleSheet](http://msdn.microsoft.com/en-us/library/ms535871(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535871\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535871(VS.85).aspx)] object associated with a class.

Tip 11: Directly modify the **style** object when changing your content presentation.

Collapse a Text Range Before Finding the Parent

A [TextRange](http://msdn.microsoft.com/en-us/library/ms535872(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535872\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535872(VS.85).aspx)] object represents an area of text that has been selected by the user or retrieved from an HTML element such as the [body](http://msdn.microsoft.com/en-us/library/ms535205(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535205\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535205(VS.85).aspx)]. The parent of the text range can be identified by calling the [parentElement](http://msdn.microsoft.com/en-us/library/ms536654(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536654\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536654(VS.85).aspx)] method. For complex text ranges, it is more efficient to call the [collapse](http://msdn.microsoft.com/en-us/library/ms536371(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms536371\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536371(VS.85).aspx)] method before calling the **parentElement** method.

Tip 12: Collapse a text range before accessing the **parentElement** property.

For more information on the TextRange object, please see [Text Range Element](http://msdn.microsoft.com/en-us/library/ms528439(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms528439\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms528439(VS.85).aspx)].

Related Topics

- [Building High Performance HTML Pages](http://msdn.microsoft.com/en-us/library/ms533020(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms533020\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533020(VS.85).aspx)]
- [More Performance Tips](http://msdn.microsoft.com/en-us/library/Bb263997(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/Bb263997\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/Bb263997(VS.85).aspx)]
- [Frequent Flyers: Boosting Performance on DHTML Pages](http://msdn.microsoft.com/en-us/library/Bb264005(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/Bb264005\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/Bb264005(VS.85).aspx)]

Tags:



Community Content