Minify Javascript with C#

September 18, 2015 amit ASP.NET, Javascript/Jquery 0

The code below compresses and minifies javascript files. Minification is not the same as obfuscation. Minification removes all whitespace , comments and needless characters so that the end result is much smaller and is more difficult to read . Obfuscation does minification as well as mangles variables and functions so that the code is completely unreadable.

The C# class takes in a javascript file and minifies and writes it out to another file.
The class can be called in a single line of code :

```
Minify js = new Minify(@"c:\myfiles\test.js");
```

The modified file will be saved as *c:\myfiles\test.js.min*

```
001 using System;
002 using System.Collections.Generic;
003 using System.Linq;
004 using System.Text;
005 using System.Threading.Tasks;
006 using System.IO;
007
008 namespace JSMinify
009 {
010     public class Minify
011     {
012         private string mFileName = "";
    // file to process
013         private string mOriginalData =
    "";           // data from original file
014         private string mModifiedData = "";
    // processed data
015         private bool mIsError =
    false;               // becomes true if any error
    happens
016         private string mErr =
    "";                  // error message
017         private BinaryReader mReader =
    null;         // stream to process the file byte by
    byte
018
019         private const int EOF = -1;
    // for end of file
020
021         /// <summary>
022         /// Constructor - does all the processing
023         /// </summary>
024         /// <param name="f">file path</param>
025         public Minify(string f) {
026
027             try
028             {
029                 if (File.Exists(f))
030                 {
031                     mFileName = f;
032
033                     //read contents completely.
    This is only for test purposes. The actual
    processing is done by another stream
034                     StreamReader rdr = new
    StreamReader(mFileName);
035                     mOriginalData =
    rdr.ReadToEnd();
036                     rdr.Close();
037
038                     mReader = new BinaryReader(new
```

Last Name:

Your last name

Email address:

Your email address

SIGN UP

CATEGORIES

Apache/PHP (47)
Programming (127)
YouTube API With PHP (31)
Python (25)
Javascript/Jquery (23)
ASP.NET (11)
C/C++ (8)
MySQL (2)
Java (2)
VC++ (1)
MongoDB (1)
Node.JS (1)
Objective C (1)
C# (1)
FreeBSD Unix (23)
Linux (21)
Misc (19)
Tutorials (18)
Windows (16)
WWW Stuff (11)
Android Dev (11)
Amazing Technology (11)
Useful Lists (9)
Xcode & Swift (6)
Web Designing (5)
Uncategorized (4)
Maps (4)
MS SQL Server (4)
MySQL (4)
Slideshow (4)
HTML (3)
Audio (3)
NginX (2)

```
FileStream(mFileName, FileMode.Open));
039                     doProcess();
040                     mReader.Close();
041 
042                     //write modified data
043                     string outFile = mFileName +
".min";
044                     StreamWriter wrt = new
StreamWriter(outFile);
045                     wrt.Write(mModifiedData);
046                     wrt.Close();
047 
048                 }
049                 else {
050                     mIsError = true;
051                     mErr = "File does not exist";
052                 }
053 
054             }
055             catch (Exception ex) {
056                 mIsError = true;
057                 mErr = ex.Message;
058             }
059         }
060 
061         /// <summary>
062         /// Main process
063         /// </summary>
064         private void doProcess() {
065             int lastChar = 1;                   //
current byte read
066             int thisChar = -1;                  //
previous byte read
067             int nextChar = -1;                  //
byte read in peek()
068             bool endProcess = false;            //
loop control
069             bool ignore = false;                //
if false then add byte to final output
070             bool inComment = false;             //
true when current bytes are part of a comment
071             bool isDoubleSlashComment = false;  //
'//' comment
072 
073 
074             // main processing loop
075             while (!endProcess) {
076                 endProcess = (mReader.PeekChar() ==
-1);    // check for EOF before reading
077                 if (endProcess)
078                     break;
079 
080                 ignore = false;
081                 thisChar = mReader.ReadByte();
082 
083                 if (thisChar == '\t')
084                     thisChar = ' ';
085                 else if (thisChar == '\t')
086                     thisChar = '\n';
087                 else if (thisChar == '\r')
088                     thisChar = '\n';
089 
090                 if (thisChar == '\n')
091                     ignore = true;
092 
```

```
093                 if (thisChar == ' ')
094                 {
095                     if ((lastChar == ' ') ||
    isDelimiter(lastChar) == 1)
096                         ignore = true;
097                     else {
098                         endProcess =
    (mReader.PeekChar() == -1); // check for EOF
099                         if (!endProcess)
100                         {
101                             nextChar =
    mReader.PeekChar();
102                             if
    (isDelimiter(nextChar) == 1)
103                                 ignore = true;
104                         }
105                     }
106                 }
107 
108 
109                 if (thisChar == '/')
110                 {
111                     nextChar = mReader.PeekChar();
112                     if (nextChar == '/' || nextChar
    == '*')
113                     {
114                         ignore = true;
115                         inComment = true;
116                         if (nextChar == '/')
117                             isDoubleSlashComment =
    true;
118                         else
119                             isDoubleSlashComment =
    false;
120                     }
121 
122 
123                 }
124 
125                 // ignore all characters till we
    reach end of comment
126                 if (inComment) {
127                     while (true) {
128                         thisChar =
    mReader.ReadByte();
129                         if (thisChar == '*') {
130                             nextChar =
    mReader.PeekChar();
131                             if (nextChar == '/')
132                             {
133                                 thisChar =
    mReader.ReadByte();
134                                 inComment = false;
135                                 break;
136                             }
137                         }
138                         if (isDoubleSlashComment &&
    thisChar == '\n') {
139                             inComment = false;
140                             break;
141                         }
142 
143                      } // while (true)
144                     ignore = true;
145                 } // if (inComment)
```

```
146
147
148                 if (!ignore)
149                     addToOutput(thisChar);
150
151                 lastChar = thisChar;
152             } // while (!endProcess)
153         }
154
155
156         /// <summary>
157         /// Add character to modified data string
158         /// </summary>
159         /// <param name="c">char to add</param>
160         private void addToOutput(int c)
161         {
162             mModifiedData += (char) c;
163         }
164
165
166         /// <summary>
167         /// Original data from file
168         /// </summary>
169         /// <returns></returns>
170         public string getOriginalData()
171         {
172             return mOriginalData;
173         }
174
175         /// <summary>
176         /// Modified data after processing
177         /// </summary>
178         /// <returns></returns>
179         public string getModifiedData()
180         {
181             return mModifiedData;
182         }
183
184         /// <summary>
185         /// Check if a byte is alphanumeric
186         /// </summary>
187         /// <param name="c">byte to check</param>
188         /// <returns>retval - 1 if yes. else
    0</returns>
189         private int isAlphanumeric(int c)
190         {
191             int retval = 0;
192
193             if ((c >= 'a' && c <= 'z') ||
194                 (c >= '0' && c <= '9') ||
195                 (c >= 'A' && c <= 'Z') ||
196                 c == '_' || c == '$' || c == '\\'
    || c > 126)
197                 retval = 1;
198
199             return retval;
200
201         }
202
203         /// <summary>
204         /// Check if a byte is a delimiter
205         /// </summary>
206         /// <param name="c">byte to check</param>
207         /// <returns>retval - 1 if yes. else
    0</returns>
```

```
        private int isDelimiter(int c)
        {
            int retval = 0;

            if (c == '(' || c == ',' || c == '=' ||
c == ':' ||
                c == '[' || c == '!' || c == '&' ||
c == '|' ||
                c == '?' || c == '+' || c == '-' ||
c == '~' ||
                c == '*' || c == '/' || c == '{' ||
c == '\n' ||
                c == ','
            )
            {
                retval = 1;
            }

            return retval;

        }



    }
}
```

An example is given below:

Original file:

```
/* jshint define: false */

/**
 * @file This plugin adds on primitive Object (like
string, number, array ...) additionnals methods
 * @version 1.0
 * @author Julien Roche
 * @copyright MIT
 */

(function(){
    "use strict";

    function definition($){
        /* String part */
        String.prototype.endWith = function
(needle) {
            return this && this.match(needle + "$")
== needle;
        };

        String.prototype.repeat = function (num) {
//          return new Array(num + 1).join(this);
            var arr = [];
            arr.length = num + 1;
            return arr.join(this);
        };

        String.prototype.startWith = function
(needle) {
            return this && this.match("^" + needle)
== needle;
```

```
028         };
029
030         /* Number part */
031         Number.prototype.toPaddedString = function
    (length, radix) {
032             var string = this.toString(radix ||
    10), slength = string.length;
033             for (var i = 0; i < (length - slength);
    i++) {
034                 string = "0" + string;
035             }
036             return string;
037         };
038
039         /* Array part */
040
041         // See http://www.to-
    string.com/2012/05/29/fixing-splice-in-older-
    versions-of-internet-explorer-8-and-olders/
042         if (document.documentMode &&
    document.documentMode < 9) {
043             // save original function of splice
044             var originalSplice =
    Array.prototype.splice;
045
046             // provide a new implementation
047             Array.prototype.splice = function() {
048
049                 // since we can't modify
    'arguments' array,
050                 // let's create a new one and copy
    all elements of 'arguments' into it
051                 var arr = [],
052                     i = 0,
053                     max = arguments.length;
054
055                 for (; i < max; i++){
056                     arr.push(arguments[i]);
057                 }
058
059                 // if this function had only one
    argument
060                 // compute 'deleteCount' and push
    it into arr
061                 if (arr.length==1) {
062                     arr.push(this.length - arr[0]);
063                 }
064
065                 // invoke original splice() with
    our new arguments array
066                 return originalSplice.apply(this,
    arr);
067             };
068         }
069
070         // See https://developer.mozilla.org/en-
    US/docs/JavaScript/Reference/Global_Objects/Array/fo
071         if(!Array.prototype.forEach) {
072             Array.prototype.forEach = function
    forEach(callback, thisArg) {
073                 var T, k;
074
075                 if(this == null) {
076                     throw new TypeError("this is
    null or not defined");
```

```
077                 }
078
079                 // 1. Let O be the result of
    calling ToObject passing the |this| value as the
    argument.
080                 var O = Object(this);
081
082                 // 2. Let lenValue be the result of
    calling the Get internal method of O with the
    argument "length".
083                 // 3. Let len be
    ToUint32(lenValue).
084                 var len = O.length >>> 0;
085                 // Hack to convert O.length to a
    UInt32
086
087                 // 4. If IsCallable(callback) is
    false, throw a TypeError exception.
088                 // See:
    http://es5.github.com/#x9.11
089                 if( {}.toString.call(callback) !==
    "[object Function]") {
090                     throw new TypeError(callback +
    " is not a function");
091                 }
092
093                 // 5. If thisArg was supplied, let
    T be thisArg; else let T be undefined.
094                 if(thisArg) {
095                     T = thisArg;
096                 }
097
098                 // 6. Let k be 0
099                 k = 0;
100
101                 // 7. Repeat, while k < len
102                 while(k < len) {
103
104                     var kValue;
105
106                     // a. Let Pk be ToString(k).
107                     //   This is implicit for LHS
    operands of the in operator
108                     // b. Let kPresent be the
    result of calling the HasProperty internal method
    of O with argument Pk.
109                     //   This step can be combined
    with c
110                     // c. If kPresent is true, then
111                     if(Object.prototype.hasOwnProper
    k)) {
112
113                         // i. Let kValue be the
    result of calling the Get internal method of O with
    argument Pk.
114                         kValue = O[k];
115
116                         // ii. Call the Call
    internal method of callback with T as the this
    value and
117                         // argument list containing
    kValue, k, and O.
118                         callback.call(T, kValue, k,
    O);
119                     }
```

```
                    // d. Increase k by 1.
                    k++;
                }
                // 8. return undefined
            };
        }
    }

    if (typeof module === "object" && typeof module.exports === "object") {
        // Node approach
        definition();

    } else if (typeof define === "function" && define.amd) {
        // AMD approach
        define("prototype", [], definition);

    } else if (window.jQuery) {
        // Classical way
        definition();
    }
}());
```

Minified file:

```
(function(){"use strict";function definition($){Stri
{return this&&this.match(needle+"$")==needle;};Strin
[];arr.length=num+1;return arr.join(this);};String.p
{return
this&&this.match("^"+needle)==needle;};Number.protot
{var string=this.toString(radix||10),slength=string.
i++){string="0"+string;} return string;};if(document
{var originalSplice=Array.prototype.splice;Array.pro
[],i=0,max=arguments.length;for(; i < max; i++){arr.
{arr.push(this.length-arr[0]);}return
originalSplice.apply(this,arr);};}if(!Array.prototyp
{Array.prototype.forEach=function forEach(callback,t
TypeError("this is null or not defined");}var O=Obje
0;if({}.toString.call(callback)!=="[object Function]
a function");}if(thisArg){T=thisArg;}k=0;while(k < l
kValue;if(Object.prototype.hasOwnProperty.call(O,k))
{kValue=O[k];callback.call(T,kValue,k,O);}k++;}};}}i
module.exports==="object"){definition();} else if(ty
{define("prototype",[],definition);} else if(window.
```

Related Posts

Minify CSS with C#

Minify Javascript with Python

Linked List class in Javascript

Adding IP restriction to a web page in ASP.NET

Visual Studio – Edit Menu Tips & Tricks

Get all HTTP headers in Javascript

« **PREVIOUS**

Capture & Send System Info by Email in Windows Powershell

NEXT »

Minify Javascript with Python

BE THE FIRST TO COMMENT

Leave a Reply

Your email address will not be published.

Comment

Name *

Email *

Website

POST COMMENT

This site uses Akismet to reduce spam. Learn how your comment data is processed.