/\*!

JSZipUtils - A collection of cross-browser utilities to go along with JSZip.

<http://stuk.github.io/jszip-utils>

(c) 2014 Stuart Knightley, David Duponchel

Dual licenced under the MIT license or GPLv3. See https://raw.github.com/Stuk/jszip-utils/master/LICENSE.markdown.

\*/

;(function e(t,n,r){function s(o,u){if(!n[o]){if(!t[o]){var a=typeof require=="function"&&require;if(!u&&a)return a(o,!0);if(i)return i(o,!0);throw new Error("Cannot find module '"+o+"'")}var f=n[o]={exports:{}};t[o][0].call(f.exports,function(e){var n=t[o][1][e];return s(n?n:e)},f,f.exports,e,t,n,r)}return n[o].exports}var i=typeof require=="function"&&require;for(var o=0;o<r.length;o++)s(r[o]);return s})({1:[function(require,module,exports){

var global=typeof self !== "undefined" ? self : typeof window !== "undefined" ? window : {};/\* jshint evil: true, newcap: false \*/

/\* global IEBinaryToArray\_ByteStr, IEBinaryToArray\_ByteStr\_Last \*/

"use strict";

// Adapted from http://stackoverflow.com/questions/1095102/how-do-i-load-binary-image-data-using-javascript-and-xmlhttprequest

var IEBinaryToArray\_ByteStr\_Script =

"<!-- IEBinaryToArray\_ByteStr -->\r\n"+

"<script type='text/vbscript'>\r\n"+

"Function IEBinaryToArray\_ByteStr(Binary)\r\n"+

" IEBinaryToArray\_ByteStr = CStr(Binary)\r\n"+

"End Function\r\n"+

"Function IEBinaryToArray\_ByteStr\_Last(Binary)\r\n"+

" Dim lastIndex\r\n"+

" lastIndex = LenB(Binary)\r\n"+

" if lastIndex mod 2 Then\r\n"+

" IEBinaryToArray\_ByteStr\_Last = Chr( AscB( MidB( Binary, lastIndex, 1 ) ) )\r\n"+

" Else\r\n"+

" IEBinaryToArray\_ByteStr\_Last = "+'""'+"\r\n"+

" End If\r\n"+

"End Function\r\n"+

"</script>\r\n";

// inject VBScript

document.write(IEBinaryToArray\_ByteStr\_Script);

global.JSZipUtils.\_getBinaryFromXHR = function (xhr) {

var binary = xhr.responseBody;

var byteMapping = {};

for ( var i = 0; i < 256; i++ ) {

for ( var j = 0; j < 256; j++ ) {

byteMapping[ String.fromCharCode( i + (j << 8) ) ] =

String.fromCharCode(i) + String.fromCharCode(j);

}

}

var rawBytes = IEBinaryToArray\_ByteStr(binary);

var lastChr = IEBinaryToArray\_ByteStr\_Last(binary);

return rawBytes.replace(/[\s\S]/g, function( match ) {

return byteMapping[match];

}) + lastChr;

};

// enforcing Stuk's coding style

// vim: set shiftwidth=4 softtabstop=4:

},{}]},{},[1])

;