/\*!

TimelineJS

Version 1.65

Designed and built by Zach Wise at VéritéCo

This program is free software: you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

http://www.gnu.org/licenses/

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* VéritéCo JS Master

Version: 0.6

Date: June 19, 2012

Copyright 2012 VéritéCo unless part of TimelineJS,

if part of TimelineJS then it inherits TimelineJS's license.

Designed and built by Zach Wise digitalartwork.net

================================================== \*/

/\* Simple JavaScript Inheritance

By John Resig http://ejohn.org/

MIT Licensed.

================================================== \*/

(function() {

var initializing = false,

fnTest = /xyz/.test(function() {

xyz;

}) ? /\b\_super\b/: /.\*/;

// The base Class implementation (does nothing)

this.Class = function() {};

// Create a new Class that inherits from this class

Class.extend = function(prop) {

var \_super = this.prototype;

// Instantiate a base class (but only create the instance,

// don't run the init constructor)

initializing = true;

var prototype = new this();

initializing = false;

// Copy the properties over onto the new prototype

for (var name in prop) {

// Check if we're overwriting an existing function

prototype[name] = typeof prop[name] == "function" &&

typeof \_super[name] == "function" && fnTest.test(prop[name]) ?

(function(name, fn) {

return function() {

var tmp = this.\_super;

// Add a new .\_super() method that is the same method

// but on the super-class

this.\_super = \_super[name];

// The method only need to be bound temporarily, so we

// remove it when we're done executing

var ret = fn.apply(this, arguments);

this.\_super = tmp;

return ret;

};

})(name, prop[name]) :

prop[name];

}

// The dummy class constructor

function Class() {

// All construction is actually done in the init method

if (!initializing && this.init)

this.init.apply(this, arguments);

}

// Populate our constructed prototype object

Class.prototype = prototype;

// Enforce the constructor to be what we expect

Class.prototype.constructor = Class;

// And make this class extendable

Class.extend = arguments.callee;

return Class;

};

})();

/\* Access to the Global Object

access the global object without hard-coding the identifier window

================================================== \*/

var global = (function () {

return this || (1,eval)('this');

}());

/\* VMM

================================================== \*/

if (typeof VMM == 'undefined') {

/\* Main Scope Container

================================================== \*/

//var VMM = {};

var VMM = Class.extend({});

/\* Debug

================================================== \*/

VMM.debug = true;

/\* Master Config

================================================== \*/

VMM.master\_config = ({

init: function() {

return this;

},

sizes: {

api: {

width: 0,

height: 0

}

},

vp: "Pellentesque nibh felis, eleifend id, commodo in, interdum vitae, leo",

api\_keys\_master: {

flickr: "RAIvxHY4hE/Elm5cieh4X5ptMyDpj7MYIxziGxi0WGCcy1s+yr7rKQ==",

google: "jwNGnYw4hE9lmAez4ll0QD+jo6SKBJFknkopLS4FrSAuGfIwyj57AusuR0s8dAo=",

twitter: ""

},

timers: {

api: 7000

},

api: {

pushques: []

},

twitter: {

active: false,

array: [],

api\_loaded: false,

que: []

},

flickr: {

active: false,

array: [],

api\_loaded: false,

que: []

},

youtube: {

active: false,

array: [],

api\_loaded: false,

que: []

},

vimeo: {

active: false,

array: [],

api\_loaded: false,

que: []

},

googlemaps: {

active: false,

map\_active: false,

places\_active: false,

array: [],

api\_loaded: false,

que: []

},

googledocs: {

active: false,

array: [],

api\_loaded: false,

que: []

},

googleplus: {

active: false,

array: [],

api\_loaded: false,

que: []

},

wikipedia: {

active: false,

array: [],

api\_loaded: false,

que: [],

tries: 0

},

soundcloud: {

active: false,

array: [],

api\_loaded: false,

que: []

}

}).init();

//VMM.createElement(tag, value, cName, attrs, styles);

VMM.createElement = function(tag, value, cName, attrs, styles) {

var ce = "";

if (tag != null && tag != "") {

// TAG

ce += "<" + tag;

if (cName != null && cName != "") {

ce += " class='" + cName + "'";

};

if (attrs != null && attrs != "") {

ce += " " + attrs;

};

if (styles != null && styles != "") {

ce += " style='" + styles + "'";

};

ce += ">";

if (value != null && value != "") {

ce += value;

}

// CLOSE TAG

ce = ce + "</" + tag + ">";

}

return ce;

};

VMM.createMediaElement = function(media, caption, credit) {

var ce = "";

var \_valid = false;

ce += "<div class='media'>";

if (media != null && media != "") {

valid = true;

ce += "<img src='" + media + "'>";

// CREDIT

if (credit != null && credit != "") {

ce += VMM.createElement("div", credit, "credit");

}

// CAPTION

if (caption != null && caption != "") {

ce += VMM.createElement("div", caption, "caption");

}

}

ce += "</div>";

return ce;

};

// Hide URL Bar for iOS and Android by Scott Jehl

// https://gist.github.com/1183357

VMM.hideUrlBar = function () {

var win = window,

doc = win.document;

// If there's a hash, or addEventListener is undefined, stop here

if( !location.hash || !win.addEventListener ){

//scroll to 1

window.scrollTo( 0, 1 );

var scrollTop = 1,

//reset to 0 on bodyready, if needed

bodycheck = setInterval(function(){

if( doc.body ){

clearInterval( bodycheck );

scrollTop = "scrollTop" in doc.body ? doc.body.scrollTop : 1;

win.scrollTo( 0, scrollTop === 1 ? 0 : 1 );

}

}, 15 );

win.addEventListener( "load", function(){

setTimeout(function(){

//reset to hide addr bar at onload

win.scrollTo( 0, scrollTop === 1 ? 0 : 1 );

}, 0);

}, false );

}

};

}

/\* Trace (console.log)

================================================== \*/

function trace( msg ) {

if (VMM.debug) {

if (window.console) {

console.log(msg);

} else if ( typeof( jsTrace ) != 'undefined' ) {

jsTrace.send( msg );

} else {

//alert(msg);

}

}

}

/\* Array Remove - By John Resig (MIT Licensed)

http://ejohn.org/blog/javascript-array-remove/

================================================== \*/

Array.prototype.remove = function(from, to) {

var rest = this.slice((to || from) + 1 || this.length);

this.length = from < 0 ? this.length + from : from;

return this.push.apply(this, rest);

}

/\* Extending Date to include Week

================================================== \*/

Date.prototype.getWeek = function() {

var onejan = new Date(this.getFullYear(),0,1);

return Math.ceil((((this - onejan) / 86400000) + onejan.getDay()+1)/7);

}

/\* Extending Date to include Day of Year

================================================== \*/

Date.prototype.getDayOfYear = function() {

var onejan = new Date(this.getFullYear(),0,1);

return Math.ceil((this - onejan) / 86400000);

}

/\* A MORE SPECIFIC TYPEOF();

// http://rolandog.com/archives/2007/01/18/typeof-a-more-specific-typeof/

================================================== \*/

// type.of()

var is={

Null:function(a){return a===null;},

Undefined:function(a){return a===undefined;},

nt:function(a){return(a===null||a===undefined);},

Function:function(a){return(typeof(a)==="function")?a.constructor.toString().match(/Function/)!==null:false;},

String:function(a){return(typeof(a)==="string")?true:(typeof(a)==="object")?a.constructor.toString().match(/string/i)!==null:false;},

Array:function(a){return(typeof(a)==="object")?a.constructor.toString().match(/array/i)!==null||a.length!==undefined:false;},

Boolean:function(a){return(typeof(a)==="boolean")?true:(typeof(a)==="object")?a.constructor.toString().match(/boolean/i)!==null:false;},

Date:function(a){return(typeof(a)==="date")?true:(typeof(a)==="object")?a.constructor.toString().match(/date/i)!==null:false;},

HTML:function(a){return(typeof(a)==="object")?a.constructor.toString().match(/html/i)!==null:false;},

Number:function(a){return(typeof(a)==="number")?true:(typeof(a)==="object")?a.constructor.toString().match(/Number/)!==null:false;},

Object:function(a){return(typeof(a)==="object")?a.constructor.toString().match(/object/i)!==null:false;},

RegExp:function(a){return(typeof(a)==="function")?a.constructor.toString().match(/regexp/i)!==null:false;}

};

var type={

of:function(a){

for(var i in is){

if(is[i](a)){

return i.toLowerCase();

}

}

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Library.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \* LIBRARY ABSTRACTION

================================================== \*/

if(typeof VMM != 'undefined') {

VMM.smoothScrollTo = function(elem, duration, ease) {

if( typeof( jQuery ) != 'undefined' ){

var \_ease = "easein",

\_duration = 1000;

if (duration != null) {

if (duration < 1) {

\_duration = 1;

} else {

\_duration = Math.round(duration);

}

}

if (ease != null && ease != "") {

\_ease = ease;

}

if (jQuery(window).scrollTop() != VMM.Lib.offset(elem).top) {

VMM.Lib.animate('html,body', \_duration, \_ease, {scrollTop: VMM.Lib.offset(elem).top})

}

}

};

VMM.attachElement = function(element, content) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).html(content);

}

};

VMM.appendElement = function(element, content) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).append(content);

}

};

VMM.getHTML = function(element) {

var e;

if( typeof( jQuery ) != 'undefined' ){

e = jQuery(element).html();

return e;

}

};

VMM.getElement = function(element, p) {

var e;

if( typeof( jQuery ) != 'undefined' ){

if (p) {

e = jQuery(element).parent().get(0);

} else {

e = jQuery(element).get(0);

}

return e;

}

};

VMM.bindEvent = function(element, the\_handler, the\_event\_type, event\_data) {

var e;

var \_event\_type = "click";

var \_event\_data = {};

if (the\_event\_type != null && the\_event\_type != "") {

\_event\_type = the\_event\_type;

}

if (\_event\_data != null && \_event\_data != "") {

\_event\_data = event\_data;

}

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).bind(\_event\_type, \_event\_data, the\_handler);

//return e;

}

};

VMM.unbindEvent = function(element, the\_handler, the\_event\_type) {

var e;

var \_event\_type = "click";

var \_event\_data = {};

if (the\_event\_type != null && the\_event\_type != "") {

\_event\_type = the\_event\_type;

}

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).unbind(\_event\_type, the\_handler);

//return e;

}

};

VMM.fireEvent = function(element, the\_event\_type, the\_data) {

var e;

var \_event\_type = "click";

var \_data = [];

if (the\_event\_type != null && the\_event\_type != "") {

\_event\_type = the\_event\_type;

}

if (the\_data != null && the\_data != "") {

\_data = the\_data;

}

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).trigger(\_event\_type, \_data);

//return e;

}

};

VMM.getJSON = function(url, data, callback) {

if( typeof( jQuery ) != 'undefined' ){

jQuery.ajaxSetup({

timeout: 3000

});

/\* CHECK FOR IE

================================================== \*/

if ( VMM.Browser.browser == "Explorer" && parseInt(VMM.Browser.version, 10) >= 7 && window.XDomainRequest) {

trace("IE JSON");

var ie\_url = url;

if (ie\_url.match('^http://')){

return jQuery.getJSON(ie\_url, data, callback);

} else if (ie\_url.match('^https://')) {

ie\_url = ie\_url.replace("https://","http://");

return jQuery.getJSON(ie\_url, data, callback);

} else {

return jQuery.getJSON(url, data, callback);

}

} else {

return jQuery.getJSON(url, data, callback);

}

}

}

VMM.parseJSON = function(the\_json) {

if( typeof( jQuery ) != 'undefined' ){

return jQuery.parseJSON(the\_json);

}

}

// ADD ELEMENT AND RETURN IT

VMM.appendAndGetElement = function(append\_to\_element, tag, cName, content) {

var e,

\_tag = "<div>",

\_class = "",

\_content = "",

\_id = "";

if (tag != null && tag != "") {

\_tag = tag;

}

if (cName != null && cName != "") {

\_class = cName;

}

if (content != null && content != "") {

\_content = content;

}

if( typeof( jQuery ) != 'undefined' ){

e = jQuery(tag);

e.addClass(\_class);

e.html(\_content);

jQuery(append\_to\_element).append(e);

}

return e;

};

VMM.Lib = {

init: function() {

return this;

},

hide: function(element, duration) {

if (duration != null && duration != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).hide(duration);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).hide();

}

}

},

remove: function(element) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).remove();

}

},

detach: function(element) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).detach();

}

},

append: function(element, value) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).append(value);

}

},

prepend: function(element, value) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).prepend(value);

}

},

show: function(element, duration) {

if (duration != null && duration != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).show(duration);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).show();

}

}

},

load: function(element, callback\_function, event\_data) {

var \_event\_data = {elem:element}; // return element by default

if (\_event\_data != null && \_event\_data != "") {

\_event\_data = event\_data;

}

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).load(\_event\_data, callback\_function);

}

},

addClass: function(element, cName) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).addClass(cName);

}

},

removeClass: function(element, cName) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).removeClass(cName);

}

},

attr: function(element, aName, value) {

if (value != null && value != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).attr(aName, value);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).attr(aName);

}

}

},

prop: function(element, aName, value) {

if (typeof jQuery == 'undefined' || !/[1-9]\.[3-9].[1-9]/.test(jQuery.fn.jquery)) {

VMM.Lib.attribute(element, aName, value);

} else {

jQuery(element).prop(aName, value);

}

},

attribute: function(element, aName, value) {

if (value != null && value != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).attr(aName, value);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).attr(aName);

}

}

},

visible: function(element, show) {

if (show != null) {

if( typeof( jQuery ) != 'undefined' ){

if (show) {

jQuery(element).show(0);

} else {

jQuery(element).hide(0);

}

}

} else {

if( typeof( jQuery ) != 'undefined' ){

if ( jQuery(element).is(':visible')){

return true;

} else {

return false;

}

}

}

},

css: function(element, prop, value) {

if (value != null && value != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).css(prop, value);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).css(prop);

}

}

},

cssmultiple: function(element, propval) {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).css(propval);

}

},

offset: function(element) {

var p;

if( typeof( jQuery ) != 'undefined' ){

p = jQuery(element).offset();

}

return p;

},

position: function(element) {

var p;

if( typeof( jQuery ) != 'undefined' ){

p = jQuery(element).position();

}

return p;

},

width: function(element, s) {

if (s != null && s != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).width(s);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).width();

}

}

},

height: function(element, s) {

if (s != null && s != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).height(s);

}

} else {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).height();

}

}

},

toggleClass: function(element, cName) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).toggleClass(cName);

}

},

each:function(element, return\_function) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).each(return\_function);

}

},

html: function(element, str) {

var e;

if( typeof( jQuery ) != 'undefined' ){

e = jQuery(element).html();

return e;

}

if (str != null && str != "") {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).html(str);

}

} else {

var e;

if( typeof( jQuery ) != 'undefined' ){

e = jQuery(element).html();

return e;

}

}

},

find: function(element, selec) {

if( typeof( jQuery ) != 'undefined' ){

return jQuery(element).find(selec);

}

},

stop: function(element) {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).stop();

}

},

delay\_animate: function(delay, element, duration, ease, att, callback\_function) {

if (VMM.Browser.device == "mobile" || VMM.Browser.device == "tablet") {

var \_tdd = Math.round((duration/1500)\*10)/10,

\_\_duration = \_tdd + 's';

VMM.Lib.css(element, '-webkit-transition', 'all '+ \_\_duration + ' ease');

VMM.Lib.css(element, '-moz-transition', 'all '+ \_\_duration + ' ease');

VMM.Lib.css(element, '-o-transition', 'all '+ \_\_duration + ' ease');

VMM.Lib.css(element, '-ms-transition', 'all '+ \_\_duration + ' ease');

VMM.Lib.css(element, 'transition', 'all '+ \_\_duration + ' ease');

VMM.Lib.cssmultiple(element, \_att);

} else {

if( typeof( jQuery ) != 'undefined' ){

jQuery(element).delay(delay).animate(att, {duration:duration, easing:ease} );

}

}

},

animate: function(element, duration, ease, att, que, callback\_function) {

var \_ease = "easein",

\_que = false,

\_duration = 1000,

\_att = {};

if (duration != null) {

if (duration < 1) {

\_duration = 1;

} else {

\_duration = Math.round(duration);

}

}

if (ease != null && ease != "") {

\_ease = ease;

}

if (que != null && que != "") {

\_que = que;

}

if (att != null) {

\_att = att

} else {

\_att = {opacity: 0}

}

if (VMM.Browser.device == "mobile" || VMM.Browser.device == "tablet") {

var \_tdd = Math.round((\_duration/1500)\*10)/10,

\_\_duration = \_tdd + 's';

\_ease = " cubic-bezier(0.33, 0.66, 0.66, 1)";

//\_ease = " ease-in-out";

for (x in \_att) {

if (Object.prototype.hasOwnProperty.call(\_att, x)) {

trace(x + " to " + \_att[x]);

VMM.Lib.css(element, '-webkit-transition', x + ' ' + \_\_duration + \_ease);

VMM.Lib.css(element, '-moz-transition', x + ' ' + \_\_duration + \_ease);

VMM.Lib.css(element, '-o-transition', x + ' ' + \_\_duration + \_ease);

VMM.Lib.css(element, '-ms-transition', x + ' ' + \_\_duration + \_ease);

VMM.Lib.css(element, 'transition', x + ' ' + \_\_duration + \_ease);

}

}

VMM.Lib.cssmultiple(element, \_att);

} else {

if( typeof( jQuery ) != 'undefined' ){

if (callback\_function != null && callback\_function != "") {

jQuery(element).animate(\_att, {queue:\_que, duration:\_duration, easing:\_ease, complete:callback\_function} );

} else {

jQuery(element).animate(\_att, {queue:\_que, duration:\_duration, easing:\_ease} );

}

}

}

}

}

}

if( typeof( jQuery ) != 'undefined' ){

/\* XDR AJAX EXTENTION FOR jQuery

https://github.com/jaubourg/ajaxHooks/blob/master/src/ajax/xdr.js

================================================== \*/

(function( jQuery ) {

if ( window.XDomainRequest ) {

jQuery.ajaxTransport(function( s ) {

if ( s.crossDomain && s.async ) {

if ( s.timeout ) {

s.xdrTimeout = s.timeout;

delete s.timeout;

}

var xdr;

return {

send: function( \_, complete ) {

function callback( status, statusText, responses, responseHeaders ) {

xdr.onload = xdr.onerror = xdr.ontimeout = jQuery.noop;

xdr = undefined;

complete( status, statusText, responses, responseHeaders );

}

xdr = new XDomainRequest();

xdr.open( s.type, s.url );

xdr.onload = function() {

callback( 200, "OK", { text: xdr.responseText }, "Content-Type: " + xdr.contentType );

};

xdr.onerror = function() {

callback( 404, "Not Found" );

};

if ( s.xdrTimeout ) {

xdr.ontimeout = function() {

callback( 0, "timeout" );

};

xdr.timeout = s.xdrTimeout;

}

xdr.send( ( s.hasContent && s.data ) || null );

},

abort: function() {

if ( xdr ) {

xdr.onerror = jQuery.noop();

xdr.abort();

}

}

};

}

});

}

})( jQuery );

/\* jQuery Easing v1.3

http://gsgd.co.uk/sandbox/jquery/easing/

================================================== \*/

jQuery.easing['jswing'] = jQuery.easing['swing'];

jQuery.extend( jQuery.easing, {

def: 'easeOutQuad',

swing: function (x, t, b, c, d) {

//alert(jQuery.easing.default);

return jQuery.easing[jQuery.easing.def](x, t, b, c, d);

},

easeInExpo: function (x, t, b, c, d) {

return (t==0) ? b : c \* Math.pow(2, 10 \* (t/d - 1)) + b;

},

easeOutExpo: function (x, t, b, c, d) {

return (t==d) ? b+c : c \* (-Math.pow(2, -10 \* t/d) + 1) + b;

},

easeInOutExpo: function (x, t, b, c, d) {

if (t==0) return b;

if (t==d) return b+c;

if ((t/=d/2) < 1) return c/2 \* Math.pow(2, 10 \* (t - 1)) + b;

return c/2 \* (-Math.pow(2, -10 \* --t) + 2) + b;

},

easeInQuad: function (x, t, b, c, d) {

return c\*(t/=d)\*t + b;

},

easeOutQuad: function (x, t, b, c, d) {

return -c \*(t/=d)\*(t-2) + b;

},

easeInOutQuad: function (x, t, b, c, d) {

if ((t/=d/2) < 1) return c/2\*t\*t + b;

return -c/2 \* ((--t)\*(t-2) - 1) + b;

}

});

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Browser.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \* DEVICE AND BROWSER DETECTION

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Browser == 'undefined') {

VMM.Browser = {

init: function () {

this.browser = this.searchString(this.dataBrowser) || "An unknown browser";

this.version = this.searchVersion(navigator.userAgent)

|| this.searchVersion(navigator.appVersion)

|| "an unknown version";

this.OS = this.searchString(this.dataOS) || "an unknown OS";

this.device = this.searchDevice(navigator.userAgent);

this.orientation = this.searchOrientation(window.orientation);

},

searchOrientation: function(orientation) {

var orient = "";

if ( orientation == 0 || orientation == 180) {

orient = "portrait";

} else if ( orientation == 90 || orientation == -90) {

orient = "landscape";

} else {

orient = "normal";

}

return orient;

},

searchDevice: function(d) {

var device = "";

if (d.match(/Android/i) || d.match(/iPhone|iPod/i)) {

device = "mobile";

} else if (d.match(/iPad/i)) {

device = "tablet";

} else if (d.match(/BlackBerry/i) || d.match(/IEMobile/i)) {

device = "other mobile";

} else {

device = "desktop";

}

return device;

},

searchString: function (data) {

for (var i=0;i<data.length;i++) {

var dataString = data[i].string,

dataProp = data[i].prop;

this.versionSearchString = data[i].versionSearch || data[i].identity;

if (dataString) {

if (dataString.indexOf(data[i].subString) != -1) {

return data[i].identity;

}

} else if (dataProp) {

return data[i].identity;

}

}

},

searchVersion: function (dataString) {

var index = dataString.indexOf(this.versionSearchString);

if (index == -1) return;

return parseFloat(dataString.substring(index+this.versionSearchString.length+1));

},

dataBrowser: [

{

string: navigator.userAgent,

subString: "Chrome",

identity: "Chrome"

},

{ string: navigator.userAgent,

subString: "OmniWeb",

versionSearch: "OmniWeb/",

identity: "OmniWeb"

},

{

string: navigator.vendor,

subString: "Apple",

identity: "Safari",

versionSearch: "Version"

},

{

prop: window.opera,

identity: "Opera",

versionSearch: "Version"

},

{

string: navigator.vendor,

subString: "iCab",

identity: "iCab"

},

{

string: navigator.vendor,

subString: "KDE",

identity: "Konqueror"

},

{

string: navigator.userAgent,

subString: "Firefox",

identity: "Firefox"

},

{

string: navigator.vendor,

subString: "Camino",

identity: "Camino"

},

{ // for newer Netscapes (6+)

string: navigator.userAgent,

subString: "Netscape",

identity: "Netscape"

},

{

string: navigator.userAgent,

subString: "MSIE",

identity: "Explorer",

versionSearch: "MSIE"

},

{

string: navigator.userAgent,

subString: "Gecko",

identity: "Mozilla",

versionSearch: "rv"

},

{ // for older Netscapes (4-)

string: navigator.userAgent,

subString: "Mozilla",

identity: "Netscape",

versionSearch: "Mozilla"

}

],

dataOS : [

{

string: navigator.platform,

subString: "Win",

identity: "Windows"

},

{

string: navigator.platform,

subString: "Mac",

identity: "Mac"

},

{

string: navigator.userAgent,

subString: "iPhone",

identity: "iPhone/iPod"

},

{

string: navigator.userAgent,

subString: "iPad",

identity: "iPad"

},

{

string: navigator.platform,

subString: "Linux",

identity: "Linux"

}

]

}

VMM.Browser.init();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.FileExtention.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \* File Extention

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.FileExtention == 'undefined') {

VMM.FileExtention = {

googleDocType: function(url) {

var fileName = url,

fileExtension = "",

validFileExtensions = ["DOC","DOCX","XLS","XLSX","PPT","PPTX","PDF","PAGES","AI","PSD","TIFF","DXF","SVG","EPS","PS","TTF","XPS","ZIP","RAR"],

flag = false;

fileExtension = fileName.substr(fileName.length - 5, 5);

for (var i = 0; i < validFileExtensions.length; i++) {

if (fileExtension.toLowerCase().match(validFileExtensions[i].toString().toLowerCase()) || fileName.match("docs.google.com") ) {

flag = true;

}

}

return flag;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Date.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \* Utilities and Useful Functions

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Date == 'undefined') {

VMM.Date = ({

init: function() {

return this;

},

dateformats: {

year: "yyyy",

month\_short: "mmm",

month: "mmmm yyyy",

full\_short: "mmm d",

full: "mmmm d',' yyyy",

time\_no\_seconds\_short: "h:MM TT",

time\_no\_seconds\_small\_date: "h:MM TT'<br/><small>'mmmm d',' yyyy'</small>'",

full\_long: "mmm d',' yyyy 'at' hh:MM TT",

full\_long\_small\_date: "hh:MM TT'<br/><small>mmm d',' yyyy'</small>'"

},

month: ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"],

month\_abbr: ["Jan.", "Feb.", "March", "April", "May", "June", "July", "Aug.", "Sept.", "Oct.", "Nov.", "Dec."],

day: ["Sunday","Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"],

day\_abbr: ["Sun.", "Mon.", "Tues.", "Wed.", "Thurs.", "Fri.", "Sat."],

hour: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],

hour\_suffix: ["am"],

//B.C.

bc\_format: {

year: "yyyy",

month\_short: "mmm",

month: "mmmm yyyy",

full\_short: "mmm d",

full: "mmmm d',' yyyy",

time\_no\_seconds\_short: "h:MM TT",

time\_no\_seconds\_small\_date: "dddd', 'h:MM TT'<br/><small>'mmmm d',' yyyy'</small>'",

full\_long: "dddd',' mmm d',' yyyy 'at' hh:MM TT",

full\_long\_small\_date: "hh:MM TT'<br/><small>'dddd',' mmm d',' yyyy'</small>'"

},

setLanguage: function(lang) {

trace("SET DATE LANGUAGE");

VMM.Date.dateformats = lang.dateformats;

VMM.Date.month = lang.date.month;

VMM.Date.month\_abbr = lang.date.month\_abbr;

VMM.Date.day = lang.date.day;

VMM.Date.day\_abbr = lang.date.day\_abbr;

dateFormat.i18n.dayNames = lang.date.day\_abbr.concat(lang.date.day);

dateFormat.i18n.monthNames = lang.date.month\_abbr.concat(lang.date.month);

},

parse: function(d) {

"use strict";

var date,

date\_array,

time\_array,

time\_parse;

if (type.of(d) == "date") {

date = d;

} else {

date = new Date(0, 0, 1, 0, 0, 0, 0);

if ( d.match(/,/gi) ) {

date\_array = d.split(",");

for(var i = 0; i < date\_array.length; i++) {

date\_array[i] = parseInt(date\_array[i], 10);

}

if ( date\_array[0] ) { date.setFullYear( date\_array[0]); }

if ( date\_array[1] > 1 ) { date.setMonth( date\_array[1] - 1); }

if ( date\_array[2] > 1 ) { date.setDate( date\_array[2]); }

if ( date\_array[3] > 1 ) { date.setHours( date\_array[3]); }

if ( date\_array[4] > 1 ) { date.setMinutes( date\_array[4]); }

if ( date\_array[5] > 1 ) { date.setSeconds( date\_array[5]); }

if ( date\_array[6] > 1 ) { date.setMilliseconds( date\_array[6]); }

} else if (d.match("/")) {

if (d.match(" ")) {

time\_parse = d.split(" ");

if (d.match(":")) {

time\_array = time\_parse[1].split(":");

if ( time\_array[0] >= 1 ) { date.setHours( time\_array[0]); }

if ( time\_array[1] >= 1 ) { date.setMinutes( time\_array[1]); }

if ( time\_array[2] >= 1 ) { date.setSeconds( time\_array[2]); }

if ( time\_array[3] >= 1 ) { date.setMilliseconds( time\_array[3]); }

}

date\_array = time\_parse[0].split("/");

} else {

date\_array = d.split("/");

}

if ( date\_array[2] ) { date.setFullYear( date\_array[2]); }

if ( date\_array[0] > 1 ) { date.setMonth( date\_array[0] - 1); }

if ( date\_array[1] > 1 ) { date.setDate( date\_array[1]); }

} else if (d.length <= 5) {

date.setFullYear(parseInt(d, 10));

date.setMonth(0);

date.setDate(1);

date.setHours(0);

date.setMinutes(0);

date.setSeconds(0);

date.setMilliseconds(0);

} else if (d.match("T")) {

if (navigator.userAgent.match(/MSIE\s(?!9.0)/)) {

// IE 8 < Won't accept dates with a "-" in them.

time\_parse = d.split("T");

if (d.match(":")) {

time\_array = \_time\_parse[1].split(":");

if ( time\_array[0] >= 1 ) { date.setHours( time\_array[0]); }

if ( time\_array[1] >= 1 ) { date.setMinutes( time\_array[1]); }

if ( time\_array[2] >= 1 ) { date.setSeconds( time\_array[2]); }

if ( time\_array[3] >= 1 ) { date.setMilliseconds( time\_array[3]); }

}

\_d\_array = time\_parse[0].split("-");

if ( date\_array[0] ) { date.setFullYear( date\_array[0]); }

if ( date\_array[1] > 1 ) { date.setMonth( date\_array[1] - 1); }

if ( date\_array[2] > 1 ) { date.setDate( date\_array[2]); }

} else {

date = new Date(Date.parse(d));

}

} else {

date = new Date(

parseInt(d.slice(0,4), 10),

parseInt(d.slice(4,6), 10) - 1,

parseInt(d.slice(6,8), 10),

parseInt(d.slice(8,10), 10),

parseInt(d.slice(10,12), 10)

);

}

}

return date;

},

prettyDate: function(d, is\_abbr, d2) {

var \_date,

\_date2,

format,

bc\_check,

is\_pair = false,

bc\_original,

bc\_number,

bc\_string;

if (d2 != null) {

is\_pair = true;

}

if (type.of(d) == "date") {

if (d.getMonth() === 0 && d.getDate() == 1 && d.getHours() === 0 && d.getMinutes() === 0 ) {

// YEAR ONLY

format = VMM.Date.dateformats.year;

} else if (d.getDate() <= 1 && d.getHours() === 0 && d.getMinutes() === 0) {

// YEAR MONTH

if (is\_abbr) {

format = VMM.Date.dateformats.month\_short;

} else {

format = VMM.Date.dateformats.month;

}

} else if (d.getHours() === 0 && d.getMinutes() === 0) {

// YEAR MONTH DAY

if (is\_abbr) {

format = VMM.Date.dateformats.full\_short;

} else {

format = VMM.Date.dateformats.full;

}

} else if (d.getMinutes() === 0) {

// YEAR MONTH DAY HOUR

if (is\_abbr) {

format = VMM.Date.dateformats.time\_no\_seconds\_short;

} else {

format = VMM.Date.dateformats.time\_no\_seconds\_small\_date;

}

} else {

// YEAR MONTH DAY HOUR MINUTE

if (is\_abbr){

format = VMM.Date.dateformats.time\_no\_seconds\_short;

} else {

format = VMM.Date.dateformats.full\_long;

}

}

\_date = dateFormat(d, format, false);

bc\_check = \_date.split(" ");

// BC TIME SUPPORT

for(var i = 0; i < bc\_check.length; i++) {

if ( parseInt(bc\_check[i], 10) < 0 ) {

trace("YEAR IS BC");

bc\_original = bc\_check[i];

bc\_number = Math.abs( parseInt(bc\_check[i], 10) );

bc\_string = bc\_number.toString() + " B.C.";

\_date = \_date.replace(bc\_original, bc\_string);

}

}

if (is\_pair) {

\_date2 = dateFormat(d2, format);

bc\_check = \_date2.split(" ");

// BC TIME SUPPORT

for(var j = 0; j < bc\_check.length; j++) {

if ( parseInt(bc\_check[j], 10) < 0 ) {

trace("YEAR IS BC");

bc\_original = bc\_check[j];

bc\_number = Math.abs( parseInt(bc\_check[j], 10) );

bc\_string = bc\_number.toString() + " B.C.";

\_date2 = \_date2.replace(bc\_original, bc\_string);

}

}

}

} else {

trace("NOT A VALID DATE?");

trace(d);

}

if (is\_pair) {

return \_date + " &mdash; " + \_date2;

} else {

return \_date;

}

}

}).init();

/\*

\* Date Format 1.2.3

\* (c) 2007-2009 Steven Levithan <stevenlevithan.com>

\* MIT license

\*

\* Includes enhancements by Scott Trenda <scott.trenda.net>

\* and Kris Kowal <cixar.com/~kris.kowal/>

\*

\* Accepts a date, a mask, or a date and a mask.

\* Returns a formatted version of the given date.

\* The date defaults to the current date/time.

\* The mask defaults to dateFormat.masks.default.

\*/

var dateFormat = function () {

var token = /d{1,4}|m{1,4}|yy(?:yy)?|([HhMsTt])\1?|[LloSZ]|"[^"]\*"|'[^']\*'/g,

timezone = /\b(?:[PMCEA][SDP]T|(?:Pacific|Mountain|Central|Eastern|Atlantic) (?:Standard|Daylight|Prevailing) Time|(?:GMT|UTC)(?:[-+]\d{4})?)\b/g,

timezoneClip = /[^-+\dA-Z]/g,

pad = function (val, len) {

val = String(val);

len = len || 2;

while (val.length < len) val = "0" + val;

return val;

};

// Regexes and supporting functions are cached through closure

return function (date, mask, utc) {

var dF = dateFormat;

// You can't provide utc if you skip other args (use the "UTC:" mask prefix)

if (arguments.length == 1 && Object.prototype.toString.call(date) == "[object String]" && !/\d/.test(date)) {

mask = date;

date = undefined;

}

// Passing date through Date applies Date.parse, if necessary

// Caused problems in IE

// date = date ? new Date(date) : new Date;

if (isNaN(date)) {

trace("invalid date " + date);

//return "";

}

mask = String(dF.masks[mask] || mask || dF.masks["default"]);

// Allow setting the utc argument via the mask

if (mask.slice(0, 4) == "UTC:") {

mask = mask.slice(4);

utc = true;

}

var \_ = utc ? "getUTC" : "get",

d = date[\_ + "Date"](),

D = date[\_ + "Day"](),

m = date[\_ + "Month"](),

y = date[\_ + "FullYear"](),

H = date[\_ + "Hours"](),

M = date[\_ + "Minutes"](),

s = date[\_ + "Seconds"](),

L = date[\_ + "Milliseconds"](),

o = utc ? 0 : date.getTimezoneOffset(),

flags = {

d: d,

dd: pad(d),

ddd: dF.i18n.dayNames[D],

dddd: dF.i18n.dayNames[D + 7],

m: m + 1,

mm: pad(m + 1),

mmm: dF.i18n.monthNames[m],

mmmm: dF.i18n.monthNames[m + 12],

yy: String(y).slice(2),

yyyy: y,

h: H % 12 || 12,

hh: pad(H % 12 || 12),

H: H,

HH: pad(H),

M: M,

MM: pad(M),

s: s,

ss: pad(s),

l: pad(L, 3),

L: pad(L > 99 ? Math.round(L / 10) : L),

t: H < 12 ? "a" : "p",

tt: H < 12 ? "am" : "pm",

T: H < 12 ? "A" : "P",

TT: H < 12 ? "AM" : "PM",

Z: utc ? "UTC" : (String(date).match(timezone) || [""]).pop().replace(timezoneClip, ""),

o: (o > 0 ? "-" : "+") + pad(Math.floor(Math.abs(o) / 60) \* 100 + Math.abs(o) % 60, 4),

S: ["th", "st", "nd", "rd"][d % 10 > 3 ? 0 : (d % 100 - d % 10 != 10) \* d % 10]

};

return mask.replace(token, function ($0) {

return $0 in flags ? flags[$0] : $0.slice(1, $0.length - 1);

});

};

}();

// Some common format strings

dateFormat.masks = {

"default": "ddd mmm dd yyyy HH:MM:ss",

shortDate: "m/d/yy",

mediumDate: "mmm d, yyyy",

longDate: "mmmm d, yyyy",

fullDate: "dddd, mmmm d, yyyy",

shortTime: "h:MM TT",

mediumTime: "h:MM:ss TT",

longTime: "h:MM:ss TT Z",

isoDate: "yyyy-mm-dd",

isoTime: "HH:MM:ss",

isoDateTime: "yyyy-mm-dd'T'HH:MM:ss",

isoUtcDateTime: "UTC:yyyy-mm-dd'T'HH:MM:ss'Z'"

};

// Internationalization strings

dateFormat.i18n = {

dayNames: [

"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat",

"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"

],

monthNames: [

"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec",

"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"

]

};

// For convenience...

Date.prototype.format = function (mask, utc) {

return dateFormat(this, mask, utc);

};

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Util.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \* Utilities and Useful Functions

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Util == 'undefined') {

VMM.Util = ({

init: function() {

return this;

},

/\* \* CORRECT PROTOCOL (DOES NOT WORK)

================================================== \*/

correctProtocol: function(url) {

var loc = (window.parent.location.protocol).toString(),

prefix = "",

the\_url = url.split("://", 2);

if (loc.match("http")) {

prefix = loc;

} else {

prefix = "https";

}

return prefix + "://" + the\_url[1];

},

/\* \* MERGE CONFIG

================================================== \*/

mergeConfig: function(config\_main, config\_to\_merge) {

var x;

for (x in config\_to\_merge) {

if (Object.prototype.hasOwnProperty.call(config\_to\_merge, x)) {

config\_main[x] = config\_to\_merge[x];

}

}

return config\_main;

},

/\* \* GET OBJECT ATTRIBUTE BY INDEX

================================================== \*/

getObjectAttributeByIndex: function(obj, index) {

if(typeof obj != 'undefined') {

var i = 0;

for (var attr in obj){

if (index === i){

return obj[attr];

}

i++;

}

return "";

} else {

return "";

}

},

/\* \* ORDINAL

================================================== \*/

ordinal: function(n) {

return ["th","st","nd","rd"][(!( ((n%10) >3) || (Math.floor(n%100/10)==1)) ) \* (n%10)];

},

/\* \* RANDOM BETWEEN

================================================== \*/

//VMM.Util.randomBetween(1, 3)

randomBetween: function(min, max) {

return Math.floor(Math.random() \* (max - min + 1) + min);

},

/\* \* AVERAGE

\* http://jsfromhell.com/array/average

\* var x = VMM.Util.average([2, 3, 4]);

\* VMM.Util.average([2, 3, 4]).mean

================================================== \*/

average: function(a) {

var r = {mean: 0, variance: 0, deviation: 0}, t = a.length;

for(var m, s = 0, l = t; l--; s += a[l]);

for(m = r.mean = s / t, l = t, s = 0; l--; s += Math.pow(a[l] - m, 2));

return r.deviation = Math.sqrt(r.variance = s / t), r;

},

/\* \* CUSTOM SORT

================================================== \*/

customSort: function(a, b) {

var a1= a, b1= b;

if(a1== b1) return 0;

return a1> b1? 1: -1;

},

/\* \* Remove Duplicates from Array

================================================== \*/

deDupeArray: function(arr) {

var i,

len=arr.length,

out=[],

obj={};

for (i=0;i<len;i++) {

obj[arr[i]]=0;

}

for (i in obj) {

out.push(i);

}

return out;

},

/\* \* Given an int or decimal, turn that into string in $xxx,xxx.xx format.

================================================== \*/

number2money: function(n, symbol, padding) {

var symbol = (symbol !== null) ? symbol : true; // add $

var padding = (padding !== null) ? padding : false; //pad with .00

var number = VMM.Math2.floatPrecision(n,2); // rounded correctly to two digits, if decimals passed

var formatted = this.niceNumber(number);

// no decimal and padding is enabled

if (!formatted.split(/\./g)[1] && padding) formatted = formatted + ".00";

// add money sign

if (symbol) formatted = "$"+formatted;

return formatted;

},

/\* \* Returns a word count number

================================================== \*/

wordCount: function(s) {

var fullStr = s + " ";

var initial\_whitespace\_rExp = /^[^A-Za-z0-9\'\-]+/gi;

var left\_trimmedStr = fullStr.replace(initial\_whitespace\_rExp, "");

var non\_alphanumerics\_rExp = /[^A-Za-z0-9\'\-]+/gi;

var cleanedStr = left\_trimmedStr.replace(non\_alphanumerics\_rExp, " ");

var splitString = cleanedStr.split(" ");

var word\_count = splitString.length -1;

if (fullStr.length <2) {

word\_count = 0;

}

return word\_count;

},

ratio: {

fit: function(w, h, ratio\_w, ratio\_h) {

//VMM.Util.ratio.fit(w, h, ratio\_w, ratio\_h).width;

var \_fit = {width:0,height:0};

// TRY WIDTH FIRST

\_fit.width = w;

//\_fit.height = Math.round((h / ratio\_h) \* ratio\_w);

\_fit.height = Math.round((w / ratio\_w) \* ratio\_h);

if (\_fit.height > h) {

\_fit.height = h;

//\_fit.width = Math.round((w / ratio\_w) \* ratio\_h);

\_fit.width = Math.round((h / ratio\_h) \* ratio\_w);

if (\_fit.width > w) {

trace("FIT: DIDN'T FIT!!! ")

}

}

return \_fit;

},

r16\_9: function(w,h) {

//VMM.Util.ratio.r16\_9(w, h) // Returns corresponding number

if (w !== null && w !== "") {

return Math.round((h / 16) \* 9);

} else if (h !== null && h !== "") {

return Math.round((w / 9) \* 16);

}

},

r4\_3: function(w,h) {

if (w !== null && w !== "") {

return Math.round((h / 4) \* 3);

} else if (h !== null && h !== "") {

return Math.round((w / 3) \* 4);

}

}

},

doubledigit: function(n) {

return (n < 10 ? '0' : '') + n;

},

/\* \* Returns a truncated segement of a long string of between min and max words. If possible, ends on a period (otherwise goes to max).

================================================== \*/

truncateWords: function(s, min, max) {

if (!min) min = 30;

if (!max) max = min;

var initial\_whitespace\_rExp = /^[^A-Za-z0-9\'\-]+/gi;

var left\_trimmedStr = s.replace(initial\_whitespace\_rExp, "");

var words = left\_trimmedStr.split(" ");

var result = [];

min = Math.min(words.length, min);

max = Math.min(words.length, max);

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

result.push(words[i]);

}

for (var j = min; i<max; i++) {

var word = words[i];

result.push(word);

if (word.charAt(word.length-1) == '.') {

break;

}

}

return (result.join(' '));

},

/\* \* Turns plain text links into real links

================================================== \*/

linkify: function(text,targets,is\_touch) {

// http://, https://, ftp://

var urlPattern = /\b(?:https?|ftp):\/\/[a-z0-9-+&@#\/%?=~\_|!:,.;]\*[a-z0-9-+&@#\/%=~\_|]/gim;

// www. sans http:// or https://

var pseudoUrlPattern = /(^|[^\/])(www\.[\S]+(\b|$))/gim;

// Email addresses

var emailAddressPattern = /(([a-zA-Z0-9\_\-\.]+)@[a-zA-Z\_]+?(?:\.[a-zA-Z]{2,6}))+/gim;

return text

.replace(urlPattern, "<a target='\_blank' href='$&' onclick='void(0)'>$&</a>")

.replace(pseudoUrlPattern, "$1<a target='\_blank' onclick='void(0)' href='http://$2'>$2</a>")

.replace(emailAddressPattern, "<a target='\_blank' onclick='void(0)' href='mailto:$1'>$1</a>");

},

linkify\_with\_twitter: function(text,targets,is\_touch) {

// http://, https://, ftp://

var urlPattern = /\b(?:https?|ftp):\/\/[a-z0-9-+&@#\/%?=~\_|!:,.;]\*[a-z0-9-+&@#\/%=~\_|]/gim;

var url\_pattern = /(\()((?:ht|f)tps?:\/\/[a-z0-9\-.\_~!$&'()\*+,;=:\/?#[\]@%]+)(\))|(\[)((?:ht|f)tps?:\/\/[a-z0-9\-.\_~!$&'()\*+,;=:\/?#[\]@%]+)(\])|(\{)((?:ht|f)tps?:\/\/[a-z0-9\-.\_~!$&'()\*+,;=:\/?#[\]@%]+)(\})|(<|&(?:lt|#60|#x3c);)((?:ht|f)tps?:\/\/[a-z0-9\-.\_~!$&'()\*+,;=:\/?#[\]@%]+)(>|&(?:gt|#62|#x3e);)|((?:^|[^=\s'"\]])\s\*['"]?|[^=\s]\s+)(\b(?:ht|f)tps?:\/\/[a-z0-9\-.\_~!$'()\*+,;=:\/?#[\]@%]+(?:(?!&(?:gt|#0\*62|#x0\*3e);|&(?:amp|apos|quot|#0\*3[49]|#x0\*2[27]);[.!&',:?;]?(?:[^a-z0-9\-.\_~!$&'()\*+,;=:\/?#[\]@%]|$))&[a-z0-9\-.\_~!$'()\*+,;=:\/?#[\]@%]\*)\*[a-z0-9\-\_~$()\*+=\/#[\]@%])/img;

var url\_replace = '$1$4$7$10$13<a href="$2$5$8$11$14" class="hyphenate">$2$5$8$11$14</a>$3$6$9$12';

// www. sans http:// or https://

var pseudoUrlPattern = /(^|[^\/])(www\.[\S]+(\b|$))/gim;

function replaceURLWithHTMLLinks(text) {

var exp = /(\b(https?|ftp|file):\/\/([-A-Z0-9+&@#%?=~\_|!:,.;]\*)([-A-Z0-9+&@#%?\/=~\_|!:,.;]\*)[-A-Z0-9+&@#\/%=~\_|])/ig;

return text.replace(exp, "<a href='$1' target='\_blank'>$3</a>");

}

// Email addresses

var emailAddressPattern = /(([a-zA-Z0-9\_\-\.]+)@[a-zA-Z\_]+?(?:\.[a-zA-Z]{2,6}))+/gim;

//var twitterHandlePattern = /(@([\w]+))/g;

var twitterHandlePattern = /\B@([\w-]+)/gm;

var twitterSearchPattern = /(#([\w]+))/g;

return text

//.replace(urlPattern, "<a target='\_blank' href='$&' onclick='void(0)'>$&</a>")

.replace(url\_pattern, url\_replace)

.replace(pseudoUrlPattern, "$1<a target='\_blank' class='hyphenate' onclick='void(0)' href='http://$2'>$2</a>")

.replace(emailAddressPattern, "<a target='\_blank' onclick='void(0)' href='mailto:$1'>$1</a>")

.replace(twitterHandlePattern, "<a href='http://twitter.com/$1' target='\_blank' onclick='void(0)'>@$1</a>")

.replace(twitterSearchPattern, "<a href='http://twitter.com/#search?q=%23$2' target='\_blank' 'void(0)'>$1</a>");

},

linkify\_wikipedia: function(text) {

var urlPattern = /<i[^>]\*>(.\*?)<\/i>/gim;

return text

.replace(urlPattern, "<a target='\_blank' href='http://en.wikipedia.org/wiki/$&' onclick='void(0)'>$&</a>")

.replace(/<i\b[^>]\*>/gim, "")

.replace(/<\/i>/gim, "")

.replace(/<b\b[^>]\*>/gim, "")

.replace(/<\/b>/gim, "");

},

/\* \* Turns plain text links into real links

================================================== \*/

// VMM.Util.unlinkify();

unlinkify: function(text) {

if(!text) return text;

text = text.replace(/<a\b[^>]\*>/i,"");

text = text.replace(/<\/a>/i, "");

return text;

},

untagify: function(text) {

if (!text) {

return text;

}

text = text.replace(/<\s\*\w.\*?>/g,"");

return text;

},

/\* \* TK

================================================== \*/

nl2br: function(text) {

return text.replace(/(\r\n|[\r\n]|\\n|\\r)/g,"<br/>");

},

/\* \* Generate a Unique ID

================================================== \*/

// VMM.Util.unique\_ID(size);

unique\_ID: function(size) {

var getRandomNumber = function(range) {

return Math.floor(Math.random() \* range);

};

var getRandomChar = function() {

var chars = "abcdefghijklmnopqurstuvwxyzABCDEFGHIJKLMNOPQURSTUVWXYZ";

return chars.substr( getRandomNumber(62), 1 );

};

var randomID = function(size) {

var str = "";

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

str += getRandomChar();

}

return str;

};

return randomID(size);

},

/\* \* Tells you if a number is even or not

================================================== \*/

// VMM.Util.isEven(n)

isEven: function(n){

return (n%2 === 0) ? true : false;

},

/\* \* Get URL Variables

================================================== \*/

// var somestring = VMM.Util.getUrlVars(str\_url)["varname"];

getUrlVars: function(string) {

var str = string.toString();

if (str.match('&#038;')) {

str = str.replace("&#038;", "&");

} else if (str.match('&#38;')) {

str = str.replace("&#38;", "&");

} else if (str.match('&amp;')) {

str = str.replace("&amp;", "&");

}

var vars = [], hash;

var hashes = str.slice(str.indexOf('?') + 1).split('&');

for(var i = 0; i < hashes.length; i++) {

hash = hashes[i].split('=');

vars.push(hash[0]);

vars[hash[0]] = hash[1];

}

return vars;

},

/\* \* Cleans up strings to become real HTML

================================================== \*/

toHTML: function(text) {

text = this.nl2br(text);

text = this.linkify(text);

return text.replace(/\s\s/g,"&nbsp;&nbsp;");

},

/\* \* Returns text strings as CamelCase

================================================== \*/

toCamelCase: function(s,forceLowerCase) {

if(forceLowerCase !== false) forceLowerCase = true;

var sps = ((forceLowerCase) ? s.toLowerCase() : s).split(" ");

for(var i=0; i<sps.length; i++) {

sps[i] = sps[i].substr(0,1).toUpperCase() + sps[i].substr(1);

}

return sps.join(" ");

},

/\* \* Replaces dumb quote marks with smart ones

================================================== \*/

properQuotes: function(str) {

return str.replace(/\"([^\"]\*)\"/gi,"&#8220;$1&#8221;");

},

/\* \* Add Commas to numbers

================================================== \*/

niceNumber: function(nStr){

nStr += '';

x = nStr.split('.');

x1 = x[0];

x2 = x.length > 1 ? '.' + x[1] : '';

var rgx = /(\d+)(\d{3})/;

while (rgx.test(x1)) {

x1 = x1.replace(rgx, '$1' + ',' + '$2');

}

return x1 + x2;

},

/\* \* Transform text to Title Case

================================================== \*/

toTitleCase: function(t){

if ( VMM.Browser.browser == "Explorer" && parseInt(VMM.Browser.version, 10) >= 7) {

return t.replace("\_", "%20");

} else {

var \_\_TitleCase = {

\_\_smallWords: ['a', 'an', 'and', 'as', 'at', 'but','by', 'en', 'for', 'if', 'in', 'of', 'on', 'or','the', 'to', 'v[.]?', 'via', 'vs[.]?'],

init: function() {

this.\_\_smallRE = this.\_\_smallWords.join('|');

this.\_\_lowerCaseWordsRE = new RegExp('\\b(' + this.\_\_smallRE + ')\\b', 'gi');

this.\_\_firstWordRE = new RegExp('^([^a-zA-Z0-9 \\r\\n\\t]\*)(' + this.\_\_smallRE + ')\\b', 'gi');

this.\_\_lastWordRE = new RegExp('\\b(' + this.\_\_smallRE + ')([^a-zA-Z0-9 \\r\\n\\t]\*)$', 'gi');

},

toTitleCase: function(string) {

var line = '';

var split = string.split(/([:.;?!][ ]|(?:[ ]|^)["“])/);

for (var i = 0; i < split.length; ++i) {

var s = split[i];

s = s.replace(/\b([a-zA-Z][a-z.'’]\*)\b/g,this.\_\_titleCaseDottedWordReplacer);

// lowercase the list of small words

s = s.replace(this.\_\_lowerCaseWordsRE, this.\_\_lowerReplacer);

// if the first word in the title is a small word then capitalize it

s = s.replace(this.\_\_firstWordRE, this.\_\_firstToUpperCase);

// if the last word in the title is a small word, then capitalize it

s = s.replace(this.\_\_lastWordRE, this.\_\_firstToUpperCase);

line += s;

}

// special cases

line = line.replace(/ V(s?)\. /g, ' v$1. ');

line = line.replace(/(['’])S\b/g, '$1s');

line = line.replace(/\b(AT&T|Q&A)\b/ig, this.\_\_upperReplacer);

return line;

},

\_\_titleCaseDottedWordReplacer: function (w) {

return (w.match(/[a-zA-Z][.][a-zA-Z]/)) ? w : \_\_TitleCase.\_\_firstToUpperCase(w);

},

\_\_lowerReplacer: function (w) { return w.toLowerCase() },

\_\_upperReplacer: function (w) { return w.toUpperCase() },

\_\_firstToUpperCase: function (w) {

var split = w.split(/(^[^a-zA-Z0-9]\*[a-zA-Z0-9])(.\*)$/);

if (split[1]) {

split[1] = split[1].toUpperCase();

}

return split.join('');

}

};

\_\_TitleCase.init();

t = t.replace(/\_/g," ");

t = \_\_TitleCase.toTitleCase(t);

return t;

}

}

}).init();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.LoadLib.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \* LoadLib Based on LazyLoad by Ryan Grove

\* https://github.com/rgrove/lazyload/

\* Copyright (c) 2011 Ryan Grove <ryan@wonko.com>

\* All rights reserved.

\* Permission is hereby granted, free of charge, to any person obtaining a copy of

\* this software and associated documentation files (the 'Software'), to deal in

\* the Software without restriction, including without limitation the rights to

\* use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

\* the Software, and to permit persons to whom the Software is furnished to do so,

\* subject to the following conditions:

\* The above copyright notice and this permission notice shall be included in all

\* copies or substantial portions of the Software.

================================================== \*/

window.loadedJS = [];

if(typeof VMM != 'undefined' && typeof VMM.LoadLib == 'undefined') {

//VMM.LoadLib.js('http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js', onJQueryLoaded);

//VMM.LoadLib.css('http://someurl.css', onCSSLoaded);

VMM.LoadLib = (function (doc) {

var env,

head,

pending = {},

pollCount = 0,

queue = {css: [], js: []},

styleSheets = doc.styleSheets;

var loaded\_Array = [];

function isLoaded(url) {

var has\_been\_loaded = false;

for(var i=0; i<loaded\_Array.length; i++) {

if (loaded\_Array[i] == url) {

has\_been\_loaded = true;

}

}

if (!has\_been\_loaded) {

loaded\_Array.push(url);

}

return has\_been\_loaded;

}

function createNode(name, attrs) {

var node = doc.createElement(name), attr;

for (attr in attrs) {

if (attrs.hasOwnProperty(attr)) {

node.setAttribute(attr, attrs[attr]);

}

}

return node;

}

function finish(type) {

var p = pending[type],

callback,

urls;

if (p) {

callback = p.callback;

urls = p.urls;

urls.shift();

pollCount = 0;

if (!urls.length) {

callback && callback.call(p.context, p.obj);

pending[type] = null;

queue[type].length && load(type);

}

}

}

function getEnv() {

var ua = navigator.userAgent;

env = {

async: doc.createElement('script').async === true

};

(env.webkit = /AppleWebKit\//.test(ua))

|| (env.ie = /MSIE/.test(ua))

|| (env.opera = /Opera/.test(ua))

|| (env.gecko = /Gecko\//.test(ua))

|| (env.unknown = true);

}

function load(type, urls, callback, obj, context) {

var \_finish = function () { finish(type); },

isCSS = type === 'css',

nodes = [],

i, len, node, p, pendingUrls, url;

env || getEnv();

if (urls) {

urls = typeof urls === 'string' ? [urls] : urls.concat();

if (isCSS || env.async || env.gecko || env.opera) {

queue[type].push({

urls : urls,

callback: callback,

obj : obj,

context : context

});

} else {

for (i = 0, len = urls.length; i < len; ++i) {

queue[type].push({

urls : [urls[i]],

callback: i === len - 1 ? callback : null,

obj : obj,

context : context

});

}

}

}

if (pending[type] || !(p = pending[type] = queue[type].shift())) {

return;

}

head || (head = doc.head || doc.getElementsByTagName('head')[0]);

pendingUrls = p.urls;

for (i = 0, len = pendingUrls.length; i < len; ++i) {

url = pendingUrls[i];

if (isCSS) {

node = env.gecko ? createNode('style') : createNode('link', {

href: url,

rel : 'stylesheet'

});

} else {

node = createNode('script', {src: url});

node.async = false;

}

node.className = 'lazyload';

node.setAttribute('charset', 'utf-8');

if (env.ie && !isCSS) {

node.onreadystatechange = function () {

if (/loaded|complete/.test(node.readyState)) {

node.onreadystatechange = null;

\_finish();

}

};

} else if (isCSS && (env.gecko || env.webkit)) {

if (env.webkit) {

p.urls[i] = node.href;

pollWebKit();

} else {

node.innerHTML = '@import "' + url + '";';

pollGecko(node);

}

} else {

node.onload = node.onerror = \_finish;

}

nodes.push(node);

}

for (i = 0, len = nodes.length; i < len; ++i) {

head.appendChild(nodes[i]);

}

}

function pollGecko(node) {

var hasRules;

try {

hasRules = !!node.sheet.cssRules;

} catch (ex) {

pollCount += 1;

if (pollCount < 200) {

setTimeout(function () { pollGecko(node); }, 50);

} else {

hasRules && finish('css');

}

return;

}

finish('css');

}

function pollWebKit() {

var css = pending.css, i;

if (css) {

i = styleSheets.length;

while (--i >= 0) {

if (styleSheets[i].href === css.urls[0]) {

finish('css');

break;

}

}

pollCount += 1;

if (css) {

if (pollCount < 200) {

setTimeout(pollWebKit, 50);

} else {

finish('css');

}

}

}

}

return {

css: function (urls, callback, obj, context) {

if (isLoaded(urls)) {

return callback;

} else {

load('css', urls, callback, obj, context);

}

},

js: function (urls, callback, obj, context) {

if (isLoaded(urls)) {

return callback;

} else {

load('js', urls, callback, obj, context);

}

}

};

})(this.document);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.ExternalAPI.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* External API

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.ExternalAPI == 'undefined') {

VMM.ExternalAPI = {

pushQues: function() {

if (VMM.master\_config.googlemaps.active) {

VMM.ExternalAPI.googlemaps.pushQue();

}

if (VMM.master\_config.youtube.active) {

VMM.ExternalAPI.youtube.pushQue();

}

if (VMM.master\_config.soundcloud.active) {

VMM.ExternalAPI.soundcloud.pushQue();

}

if (VMM.master\_config.googledocs.active) {

VMM.ExternalAPI.googledocs.pushQue();

}

if (VMM.master\_config.googleplus.active) {

VMM.ExternalAPI.googleplus.pushQue();

}

if (VMM.master\_config.wikipedia.active) {

VMM.ExternalAPI.wikipedia.pushQue();

}

if (VMM.master\_config.vimeo.active) {

VMM.ExternalAPI.vimeo.pushQue();

}

if (VMM.master\_config.twitter.active) {

VMM.ExternalAPI.twitter.pushQue();

}

if (VMM.master\_config.flickr.active) {

VMM.ExternalAPI.flickr.pushQue();

}

},

twitter: {

tweetArray: [],

get: function(mid, id) {

var tweet = {mid: mid, id: id};

VMM.master\_config.twitter.que.push(tweet);

VMM.master\_config.twitter.active = true;

//VMM.master\_config.api.pushques.push(VMM.ExternalAPI.twitter.pushQue);

},

create: function(tweet, callback) {

var id = tweet.mid.toString(),

error\_obj = { twitterid: tweet.mid },

the\_url = "http://api.twitter.com/1/statuses/show.json?id=" + tweet.mid + "&include\_entities=true&callback=?",

twitter\_timeout = setTimeout(VMM.ExternalAPI.twitter.errorTimeOut, VMM.master\_config.timers.api, tweet),

callback\_timeout= setTimeout(callback, VMM.master\_config.timers.api, tweet);

VMM.getJSON(the\_url, function(d) {

var id = d.id\_str,

twit = "<blockquote><p>",

td = VMM.Util.linkify\_with\_twitter(d.text, "\_blank");

// TWEET CONTENT

twit += td;

twit += "</p></blockquote>";

// TWEET MEDIA

if (typeof d.entities.media != 'undefined') {

if (d.entities.media[0].type == "photo") {

//twit += "<img src=' " + d.entities.media[0].media\_url + "' alt=''>"

}

}

// TWEET AUTHOR

twit += "<div class='vcard author'>";

twit += "<a class='screen-name url' href='https://twitter.com/" + d.user.screen\_name + "' data-screen-name='" + d.user.screen\_name + "' target='\_blank'>";

twit += "<span class='avatar'><img src=' " + d.user.profile\_image\_url + "' alt=''></span>";

twit += "<span class='fn'>" + d.user.name + "</span>";

twit += "<span class='nickname'>@" + d.user.screen\_name + "<span class='thumbnail-inline'></span></span>";

twit += "</a>";

twit += "</div>";

VMM.attachElement("#"+tweet.id.toString(), twit );

VMM.attachElement("#text\_thumb\_"+tweet.id.toString(), d.text );

})

.error(function(jqXHR, textStatus, errorThrown) {

trace("TWITTER error");

trace("TWITTER ERROR: " + textStatus + " " + jqXHR.responseText);

VMM.attachElement("#"+tweet.id, VMM.MediaElement.loadingmessage("ERROR LOADING TWEET " + tweet.mid) );

})

.success(function(d) {

clearTimeout(twitter\_timeout);

clearTimeout(callback\_timeout);

callback();

});

},

errorTimeOut: function(tweet) {

trace("TWITTER JSON ERROR TIMEOUT " + tweet.mid);

VMM.attachElement("#"+tweet.id.toString(), VMM.MediaElement.loadingmessage("Still waiting on Twitter: " + tweet.mid) );

// CHECK RATE STATUS

VMM.getJSON("http://api.twitter.com/1/account/rate\_limit\_status.json", function(d) {

trace("REMAINING TWITTER API CALLS " + d.remaining\_hits);

trace("TWITTER RATE LIMIT WILL RESET AT " + d.reset\_time);

var mes = "";

if (d.remaining\_hits == 0) {

mes = "<p>You've reached the maximum number of tweets you can load in an hour.</p>";

mes += "<p>You can view tweets again starting at: <br/>" + d.reset\_time + "</p>";

} else {

mes = "<p>Still waiting on Twitter. " + tweet.mid + "</p>";

//mes = "<p>Tweet " + id + " was not found.</p>";

}

VMM.attachElement("#"+tweet.id.toString(), VMM.MediaElement.loadingmessage(mes) );

});

},

pushQue: function() {

if (VMM.master\_config.twitter.que.length > 0) {

VMM.ExternalAPI.twitter.create(VMM.master\_config.twitter.que[0], VMM.ExternalAPI.twitter.pushQue);

VMM.master\_config.twitter.que.remove(0);

}

},

getHTML: function(id) {

//var the\_url = document.location.protocol + "//api.twitter.com/1/statuses/oembed.json?id=" + id+ "&callback=?";

var the\_url = "http://api.twitter.com/1/statuses/oembed.json?id=" + id+ "&callback=?";

VMM.getJSON(the\_url, VMM.ExternalAPI.twitter.onJSONLoaded);

},

onJSONLoaded: function(d) {

trace("TWITTER JSON LOADED");

var id = d.id;

VMM.attachElement("#"+id, VMM.Util.linkify\_with\_twitter(d.html) );

},

parseTwitterDate: function(d) {

var date = new Date(Date.parse(d));

/\*

var t = d.replace(/(\d{1,2}[:]\d{2}[:]\d{2}) (.\*)/, '$2 $1');

t = t.replace(/(\+\S+) (.\*)/, '$2 $1');

var date = new Date(Date.parse(t)).toLocaleDateString();

var time = new Date(Date.parse(t)).toLocaleTimeString();

\*/

return date;

},

prettyParseTwitterDate: function(d) {

var date = new Date(Date.parse(d));

return VMM.Date.prettyDate(date, true);

},

getTweets: function(tweets) {

var tweetArray = [];

var number\_of\_tweets = tweets.length;

for(var i = 0; i < tweets.length; i++) {

var twitter\_id = "";

/\* FIND THE TWITTER ID

================================================== \*/

if (tweets[i].tweet.match("status\/")) {

twitter\_id = tweets[i].tweet.split("status\/")[1];

} else if (tweets[i].tweet.match("statuses\/")) {

twitter\_id = tweets[i].tweet.split("statuses\/")[1];

} else {

twitter\_id = "";

}

/\* FETCH THE DATA

================================================== \*/

var the\_url = "http://api.twitter.com/1/statuses/show.json?id=" + twitter\_id + "&include\_entities=true&callback=?";

VMM.getJSON(the\_url, function(d) {

var tweet = {}

/\* FORMAT RESPONSE

================================================== \*/

var twit = "<div class='twitter'><blockquote><p>";

var td = VMM.Util.linkify\_with\_twitter(d.text, "\_blank");

twit += td;

twit += "</p>";

twit += "— " + d.user.name + " (<a href='https://twitter.com/" + d.user.screen\_name + "'>@" + d.user.screen\_name + "</a>) <a href='https://twitter.com/" + d.user.screen\_name + "/status/" + d.id + "'>" + VMM.ExternalAPI.twitter.prettyParseTwitterDate(d.created\_at) + " </a></blockquote></div>";

tweet.content = twit;

tweet.raw = d;

tweetArray.push(tweet);

/\* CHECK IF THATS ALL OF THEM

================================================== \*/

if (tweetArray.length == number\_of\_tweets) {

var the\_tweets = {tweetdata: tweetArray}

VMM.fireEvent(global, "TWEETSLOADED", the\_tweets);

}

})

.success(function() { trace("second success"); })

.error(function() { trace("error"); })

.complete(function() { trace("complete"); });

}

},

getTweetSearch: function(tweets, number\_of\_tweets) {

var \_number\_of\_tweets = 40;

if (number\_of\_tweets != null && number\_of\_tweets != "") {

\_number\_of\_tweets = number\_of\_tweets;

}

var the\_url = "http://search.twitter.com/search.json?q=" + tweets + "&rpp=" + \_number\_of\_tweets + "&include\_entities=true&result\_type=mixed";

var tweetArray = [];

VMM.getJSON(the\_url, function(d) {

/\* FORMAT RESPONSE

================================================== \*/

for(var i = 0; i < d.results.length; i++) {

var tweet = {}

var twit = "<div class='twitter'><blockquote><p>";

var td = VMM.Util.linkify\_with\_twitter(d.results[i].text, "\_blank");

twit += td;

twit += "</p>";

twit += "— " + d.results[i].from\_user\_name + " (<a href='https://twitter.com/" + d.results[i].from\_user + "'>@" + d.results[i].from\_user + "</a>) <a href='https://twitter.com/" + d.results[i].from\_user + "/status/" + d.id + "'>" + VMM.ExternalAPI.twitter.prettyParseTwitterDate(d.results[i].created\_at) + " </a></blockquote></div>";

tweet.content = twit;

tweet.raw = d.results[i];

tweetArray.push(tweet);

}

var the\_tweets = {tweetdata: tweetArray}

VMM.fireEvent(global, "TWEETSLOADED", the\_tweets);

});

},

prettyHTML: function(id, secondary) {

var id = id.toString();

var error\_obj = {

twitterid: id

};

var the\_url = "http://api.twitter.com/1/statuses/show.json?id=" + id + "&include\_entities=true&callback=?";

var twitter\_timeout = setTimeout(VMM.ExternalAPI.twitter.errorTimeOut, VMM.master\_config.timers.api, id);

VMM.getJSON(the\_url, VMM.ExternalAPI.twitter.formatJSON)

.error(function(jqXHR, textStatus, errorThrown) {

trace("TWITTER error");

trace("TWITTER ERROR: " + textStatus + " " + jqXHR.responseText);

VMM.attachElement("#twitter\_"+id, "<p>ERROR LOADING TWEET " + id + "</p>" );

})

.success(function(d) {

clearTimeout(twitter\_timeout);

if (secondary) {

VMM.ExternalAPI.twitter.secondaryMedia(d);

}

});

},

formatJSON: function(d) {

var id = d.id\_str;

var twit = "<blockquote><p>";

var td = VMM.Util.linkify\_with\_twitter(d.text, "\_blank");

//td = td.replace(/(@([\w]+))/g,"<a href='http://twitter.com/$2' target='\_blank'>$1</a>");

//td = td.replace(/(#([\w]+))/g,"<a href='http://twitter.com/#search?q=%23$2' target='\_blank'>$1</a>");

twit += td;

twit += "</p></blockquote>";

//twit += " <a href='https://twitter.com/" + d.user.screen\_name + "/status/" + d.id\_str + "' target='\_blank' alt='link to original tweet' title='link to original tweet'>" + "<span class='created-at'></span>" + " </a>";

twit += "<div class='vcard author'>";

twit += "<a class='screen-name url' href='https://twitter.com/" + d.user.screen\_name + "' data-screen-name='" + d.user.screen\_name + "' target='\_blank'>";

twit += "<span class='avatar'><img src=' " + d.user.profile\_image\_url + "' alt=''></span>";

twit += "<span class='fn'>" + d.user.name + "</span>";

twit += "<span class='nickname'>@" + d.user.screen\_name + "<span class='thumbnail-inline'></span></span>";

twit += "</a>";

twit += "</div>";

if (typeof d.entities.media != 'undefined') {

if (d.entities.media[0].type == "photo") {

twit += "<img src=' " + d.entities.media[0].media\_url + "' alt=''>"

}

}

VMM.attachElement("#twitter\_"+id.toString(), twit );

VMM.attachElement("#text\_thumb\_"+id.toString(), d.text );

}

},

googlemaps: {

get: function(url, id) {

var timer, api\_key, map\_vars;

map\_vars = VMM.Util.getUrlVars(url);

if (VMM.master\_config.Timeline.api\_keys.google != "") {

api\_key = VMM.master\_config.Timeline.api\_keys.google;

} else {

api\_key = Aes.Ctr.decrypt(VMM.master\_config.api\_keys\_master.google, VMM.master\_config.vp, 256);

}

var map\_url = "http://maps.googleapis.com/maps/api/js?key=" + api\_key + "&libraries=places&sensor=false&callback=VMM.ExternalAPI.googlemaps.onMapAPIReady";

var map = { url: url, vars: map\_vars, id: id };

if (VMM.master\_config.googlemaps.active) {

VMM.master\_config.googlemaps.que.push(map);

} else {

VMM.master\_config.googlemaps.que.push(map);

if (VMM.master\_config.googlemaps.api\_loaded) {

} else {

VMM.LoadLib.js(map\_url, function() {

trace("Google Maps API Library Loaded");

});

}

}

},

create: function(m) {

var map\_attribution = "";

var layer;

var map;

function mapProvider(name) {

if (name in VMM.ExternalAPI.googlemaps.map\_providers) {

map\_attribution = VMM.ExternalAPI.googlemaps.map\_attribution[VMM.ExternalAPI.googlemaps.map\_providers[name].attribution];

return VMM.ExternalAPI.googlemaps.map\_providers[name];

} else {

if (VMM.ExternalAPI.googlemaps.defaultType(name)) {

trace("GOOGLE MAP DEFAULT TYPE");

return google.maps.MapTypeId[name.toUpperCase()];

} else {

trace("Not a maptype: " + name );

}

}

}

google.maps.VeriteMapType = function(name) {

if (VMM.ExternalAPI.googlemaps.defaultType(name)) {

return google.maps.MapTypeId[name.toUpperCase()];

} else {

var provider = mapProvider(name);

return google.maps.ImageMapType.call(this, {

"getTileUrl": function(coord, zoom) {

var index = (zoom + coord.x + coord.y) % VMM.ExternalAPI.googlemaps.map\_subdomains.length;

return [

provider.url

.replace("{S}", VMM.ExternalAPI.googlemaps.map\_subdomains[index])

.replace("{Z}", zoom)

.replace("{X}", coord.x)

.replace("{Y}", coord.y)

.replace("{z}", zoom)

.replace("{x}", coord.x)

.replace("{y}", coord.y)

];

},

"tileSize": new google.maps.Size(256, 256),

"name": name,

"minZoom": provider.minZoom,

"maxZoom": provider.maxZoom

});

}

};

google.maps.VeriteMapType.prototype = new google.maps.ImageMapType("\_");

/\* Make the Map

================================================== \*/

if (type.of(VMM.master\_config.Timeline.maptype) == "string") {

if (VMM.ExternalAPI.googlemaps.defaultType(VMM.master\_config.Timeline.maptype)) {

layer = google.maps.MapTypeId[VMM.master\_config.Timeline.maptype.toUpperCase()];

} else {

layer = VMM.master\_config.Timeline.maptype;

}

} else {

layer = "toner";

}

var location = new google.maps.LatLng(41.875696,-87.624207);

var latlong;

var zoom = 11;

var has\_location = false;

var has\_zoom = false;

var map\_bounds;

if (type.of(VMM.Util.getUrlVars(m.url)["ll"]) == "string") {

has\_location = true;

latlong = VMM.Util.getUrlVars(m.url)["ll"].split(",");

location = new google.maps.LatLng(parseFloat(latlong[0]),parseFloat(latlong[1]));

} else if (type.of(VMM.Util.getUrlVars(m.url)["sll"]) == "string") {

latlong = VMM.Util.getUrlVars(m.url)["sll"].split(",");

location = new google.maps.LatLng(parseFloat(latlong[0]),parseFloat(latlong[1]));

}

if (type.of(VMM.Util.getUrlVars(m.url)["z"]) == "string") {

has\_zoom = true;

zoom = parseFloat(VMM.Util.getUrlVars(m.url)["z"]);

}

var map\_options = {

zoom: zoom,

disableDefaultUI: true,

mapTypeControl: false,

zoomControl: true,

zoomControlOptions: {

style: google.maps.ZoomControlStyle.SMALL,

position: google.maps.ControlPosition.TOP\_RIGHT

},

center: location,

mapTypeId: layer,

mapTypeControlOptions: {

mapTypeIds: [layer]

}

}

var unique\_map\_id = m.id.toString() + "\_gmap";

VMM.attachElement("#" + m.id, "<div class='google-map' id='" + unique\_map\_id + "' style='width=100%;height=100%;'></div>");

var map = new google.maps.Map(document.getElementById(unique\_map\_id), map\_options);

if (VMM.ExternalAPI.googlemaps.defaultType(VMM.master\_config.Timeline.maptype)) {

} else {

map.mapTypes.set(layer, new google.maps.VeriteMapType(layer));

// ATTRIBUTION

var map\_attribution\_html = "<div class='map-attribution'><div class='attribution-text'>" + map\_attribution + "</div></div>";

VMM.appendElement("#"+unique\_map\_id, map\_attribution\_html);

}

loadKML();

// KML

function loadKML() {

var kml\_url = m.url + "&output=kml";

kml\_url = kml\_url.replace("&output=embed", "");

var kml\_layer = new google.maps.KmlLayer(kml\_url, {preserveViewport:true});

var infowindow = new google.maps.InfoWindow();

kml\_layer.setMap(map);

google.maps.event.addListenerOnce(kml\_layer, "defaultviewport\_changed", function() {

map.fitBounds(kml\_layer.getDefaultViewport() );

if (has\_location) {

map.panTo(location);

}

if (has\_zoom) {

map.setZoom(zoom);

}

});

google.maps.event.addListener(kml\_layer, 'click', function(kmlEvent) {

var text = kmlEvent.featureData.description;

showInfoWindow(text);

function showInfoWindow(c) {

infowindow.setContent(c);

infowindow.open(map);

}

});

}

},

pushQue: function() {

for(var i = 0; i < VMM.master\_config.googlemaps.que.length; i++) {

VMM.ExternalAPI.googlemaps.create(VMM.master\_config.googlemaps.que[i]);

}

VMM.master\_config.googlemaps.que = [];

},

onMapAPIReady: function() {

VMM.master\_config.googlemaps.map\_active = true;

VMM.master\_config.googlemaps.places\_active = true;

VMM.ExternalAPI.googlemaps.onAPIReady();

},

onPlacesAPIReady: function() {

VMM.master\_config.googlemaps.places\_active = true;

VMM.ExternalAPI.googlemaps.onAPIReady();

},

onAPIReady: function() {

if (!VMM.master\_config.googlemaps.active) {

if (VMM.master\_config.googlemaps.map\_active && VMM.master\_config.googlemaps.places\_active) {

VMM.master\_config.googlemaps.active = true;

VMM.ExternalAPI.googlemaps.pushQue();

}

}

},

defaultType: function(name) {

if (name.toLowerCase() == "satellite" || name.toLowerCase() == "hybrid" || name.toLowerCase() == "terrain" || name.toLowerCase() == "roadmap") {

return true;

} else {

return false;

}

},

map\_subdomains: ["", "a.", "b.", "c.", "d."],

map\_attribution: {

"stamen": "Map tiles by <a href='http://stamen.com'>Stamen Design</a>, under <a href='http://creativecommons.org/licenses/by/3.0'>CC BY 3.0</a>. Data by <a href='http://openstreetmap.org'>OpenStreetMap</a>, under <a href='http://creativecommons.org/licenses/by-sa/3.0'>CC BY SA</a>.",

"apple": "Map data &copy; 2012 Apple, Imagery &copy; 2012 Apple"

},

map\_providers: {

"toner": {

"url": "http://{S}tile.stamen.com/toner/{Z}/{X}/{Y}.png",

"minZoom": 0,

"maxZoom": 20,

"attribution": "stamen"

},

"toner-lines": {

"url": "http://{S}tile.stamen.com/toner-lines/{Z}/{X}/{Y}.png",

"minZoom": 0,

"maxZoom": 20,

"attribution": "stamen"

},

"toner-labels": {

"url": "http://{S}tile.stamen.com/toner-labels/{Z}/{X}/{Y}.png",

"minZoom": 0,

"maxZoom": 20,

"attribution": "stamen"

},

"sterrain": {

"url": "http://{S}tile.stamen.com/terrain/{Z}/{X}/{Y}.jpg",

"minZoom": 4,

"maxZoom": 20,

"attribution": "stamen"

},

"apple": {

"url": "http://gsp2.apple.com/tile?api=1&style=slideshow&layers=default&lang=en\_US&z={z}&x={x}&y={y}&v=9",

"minZoom": 4,

"maxZoom": 14,

"attribution": "apple"

},

"watercolor": {

"url": "http://{S}tile.stamen.com/watercolor/{Z}/{X}/{Y}.jpg",

"minZoom": 3,

"maxZoom": 16,

"attribution": "stamen"

}

}

},

googleplus: {

get: function(user, activity) {

var api\_key;

var gplus = {user: user, activity: activity};

VMM.master\_config.googleplus.que.push(gplus);

VMM.master\_config.googleplus.active = true;

},

create: function(gplus, callback) {

var mediaElem = "",

api\_key = "",

g\_activity = "",

g\_content = "",

g\_attachments = "",

gperson\_api\_url,

gactivity\_api\_url;

googleplus\_timeout = setTimeout(VMM.ExternalAPI.googleplus.errorTimeOut, VMM.master\_config.timers.api, gplus),

callback\_timeout = setTimeout(callback, VMM.master\_config.timers.api, gplus);

if (VMM.master\_config.Timeline.api\_keys.google != "") {

api\_key = VMM.master\_config.Timeline.api\_keys.google;

} else {

api\_key = Aes.Ctr.decrypt(VMM.master\_config.api\_keys\_master.google, VMM.master\_config.vp, 256);

}

gperson\_api\_url = "https://www.googleapis.com/plus/v1/people/" + gplus.user + "/activities/public?alt=json&maxResults=100&fields=items(id,url)&key=" + api\_key;

//mediaElem = "<iframe class='doc' frameborder='0' width='100%' height='100%' src='" + gplus.url + "&amp;embedded=true'></iframe>";

mediaElem = "GOOGLE PLUS API CALL";

VMM.getJSON(gperson\_api\_url, function(p\_data) {

for(var i = 0; i < p\_data.items.length; i++) {

trace("loop");

if (p\_data.items[i].url.split("posts/")[1] == gplus.activity) {

trace("FOUND IT!!");

g\_activity = p\_data.items[i].id;

gactivity\_api\_url = "https://www.googleapis.com/plus/v1/activities/" + g\_activity + "?alt=json&key=" + api\_key;

VMM.getJSON(gactivity\_api\_url, function(a\_data) {

trace(a\_data);

//a\_data.url

//a\_data.image.url

//a\_data.actor.displayName

//a\_data.provider.title

//a\_data.object.content

//g\_content += "<h4>" + a\_data.title + "</h4>";

if (typeof a\_data.annotation != 'undefined') {

g\_content += "<div class='googleplus-annotation'>'" + a\_data.annotation + "</div>";

g\_content += a\_data.object.content;

} else {

g\_content += a\_data.object.content;

}

if (typeof a\_data.object.attachments != 'undefined') {

//g\_attachments += "<div class='googleplus-attachemnts'>";

for(var k = 0; k < a\_data.object.attachments.length; k++) {

if (a\_data.object.attachments[k].objectType == "photo") {

g\_attachments = "<a href='" + a\_data.object.url + "' target='\_blank'>" + "<img src='" + a\_data.object.attachments[k].image.url + "' class='article-thumb'></a>" + g\_attachments;

} else if (a\_data.object.attachments[k].objectType == "video") {

g\_attachments = "<img src='" + a\_data.object.attachments[k].image.url + "' class='article-thumb'>" + g\_attachments;

g\_attachments += "<div>";

g\_attachments += "<a href='" + a\_data.object.attachments[k].url + "' target='\_blank'>"

g\_attachments += "<h5>" + a\_data.object.attachments[k].displayName + "</h5>";

//g\_attachments += "<p>" + a\_data.object.attachments[k].content + "</p>";

g\_attachments += "</a>";

g\_attachments += "</div>";

} else if (a\_data.object.attachments[k].objectType == "article") {

g\_attachments += "<div>";

g\_attachments += "<a href='" + a\_data.object.attachments[k].url + "' target='\_blank'>"

g\_attachments += "<h5>" + a\_data.object.attachments[k].displayName + "</h5>";

g\_attachments += "<p>" + a\_data.object.attachments[k].content + "</p>";

g\_attachments += "</a>";

g\_attachments += "</div>";

}

trace(a\_data.object.attachments[k]);

}

g\_attachments = "<div class='googleplus-attachments'>" + g\_attachments + "</div>";

}

//mediaElem = "<div class='googleplus'>";

mediaElem = "<div class='googleplus-content'>" + g\_content + g\_attachments + "</div>";

mediaElem += "<div class='vcard author'><a class='screen-name url' href='" + a\_data.url + "' target='\_blank'>";

mediaElem += "<span class='avatar'><img src='" + a\_data.actor.image.url + "' style='max-width: 32px; max-height: 32px;'></span>"

mediaElem += "<span class='fn'>" + a\_data.actor.displayName + "</span>";

mediaElem += "<span class='nickname'><span class='thumbnail-inline'></span></span>";

mediaElem += "</a></div>";

VMM.attachElement("#googleplus\_" + gplus.activity, mediaElem);

});

break;

}

}

})

.error(function(jqXHR, textStatus, errorThrown) {

var error\_obj = VMM.parseJSON(jqXHR.responseText);

trace(error\_obj.error.message);

VMM.attachElement("#googleplus\_" + gplus.activity, VMM.MediaElement.loadingmessage("<p>ERROR LOADING GOOGLE+ </p><p>" + error\_obj.error.message + "</p>"));

})

.success(function(d) {

clearTimeout(googleplus\_timeout);

clearTimeout(callback\_timeout);

callback();

});

},

pushQue: function() {

if (VMM.master\_config.googleplus.que.length > 0) {

VMM.ExternalAPI.googleplus.create(VMM.master\_config.googleplus.que[0], VMM.ExternalAPI.googleplus.pushQue);

VMM.master\_config.googleplus.que.remove(0);

}

/\*

for(var i = 0; i < VMM.master\_config.googleplus.que.length; i++) {

VMM.ExternalAPI.googleplus.create(VMM.master\_config.googleplus.que[i]);

}

VMM.master\_config.googleplus.que = [];

\*/

},

errorTimeOut: function(gplus) {

trace("GOOGLE+ JSON ERROR TIMEOUT " + gplus.activity);

VMM.attachElement("#googleplus\_" + gplus.activity, VMM.MediaElement.loadingmessage("<p>Still waiting on GOOGLE+ </p><p>" + gplus.activity + "</p>"));

}

},

googledocs: {

get: function(url, id) {

var doc = {url: url, id: id};

VMM.master\_config.googledocs.que.push(doc);

VMM.master\_config.googledocs.active = true;

},

create: function(doc) {

var mediaElem = "";

if (doc.url.match(/docs.google.com/i)) {

mediaElem = "<iframe class='doc' frameborder='0' width='100%' height='100%' src='" + doc.url + "&amp;embedded=true'></iframe>";

} else {

mediaElem = "<iframe class='doc' frameborder='0' width='100%' height='100%' src='" + "http://docs.google.com/viewer?url=" + doc.url + "&amp;embedded=true'></iframe>";

}

VMM.attachElement("#"+doc.id, mediaElem);

},

pushQue: function() {

for(var i = 0; i < VMM.master\_config.googledocs.que.length; i++) {

VMM.ExternalAPI.googledocs.create(VMM.master\_config.googledocs.que[i]);

}

VMM.master\_config.googledocs.que = [];

}

},

flickr: {

get: function(mid, id, link) {

var flick = {mid: mid, id: id, link:link};

VMM.master\_config.flickr.que.push(flick);

VMM.master\_config.flickr.active = true;

},

create: function(flick, callback) {

var api\_key,

callback\_timeout= setTimeout(callback, VMM.master\_config.timers.api, flick);

if (VMM.master\_config.Timeline.api\_keys.flickr != "") {

api\_key = VMM.master\_config.Timeline.api\_keys.flickr;

} else {

api\_key = Aes.Ctr.decrypt(VMM.master\_config.api\_keys\_master.flickr, VMM.master\_config.vp, 256)

}

var the\_url = "http://api.flickr.com/services/rest/?method=flickr.photos.getSizes&api\_key=" + api\_key + "&photo\_id=" + flick.mid + "&format=json&jsoncallback=?";

VMM.getJSON(the\_url, function(d) {

var flickr\_id = d.sizes.size[0].url.split("photos\/")[1].split("/")[1];

var flickr\_large\_id = "#" + flick.id,

flickr\_thumb\_id = "#" + flick.id + "\_thumb";

//flickr\_thumb\_id = "flickr\_" + uid + "\_thumb";

var flickr\_img\_size,

flickr\_img\_thumb,

flickr\_size\_found = false,

flickr\_best\_size = "Large";

flickr\_best\_size = VMM.ExternalAPI.flickr.sizes(VMM.master\_config.sizes.api.height);

for(var i = 0; i < d.sizes.size.length; i++) {

if (d.sizes.size[i].label == flickr\_best\_size) {

flickr\_size\_found = true;

flickr\_img\_size = d.sizes.size[i].source;

}

}

if (!flickr\_size\_found) {

flickr\_img\_size = d.sizes.size[d.sizes.size.length - 1].source;

}

flickr\_img\_thumb = d.sizes.size[0].source;

VMM.Lib.attr(flickr\_large\_id, "src", flickr\_img\_size);

//VMM.attachElement(flickr\_large\_id, "<a href='" + flick.link + "' target='\_blank'><img src='" + flickr\_img\_size + "'></a>");

VMM.attachElement(flickr\_thumb\_id, "<img src='" + flickr\_img\_thumb + "'>");

})

.error(function(jqXHR, textStatus, errorThrown) {

trace("FLICKR error");

trace("FLICKR ERROR: " + textStatus + " " + jqXHR.responseText);

})

.success(function(d) {

clearTimeout(callback\_timeout);

callback();

});

},

pushQue: function() {

if (VMM.master\_config.flickr.que.length > 0) {

VMM.ExternalAPI.flickr.create(VMM.master\_config.flickr.que[0], VMM.ExternalAPI.flickr.pushQue);

VMM.master\_config.flickr.que.remove(0);

}

},

sizes: function(s) {

var \_size = "";

if (s <= 75) {

\_size = "Thumbnail";

} else if (s <= 180) {

\_size = "Small";

} else if (s <= 240) {

\_size = "Small 320";

} else if (s <= 375) {

\_size = "Medium";

} else if (s <= 480) {

\_size = "Medium 640";

} else if (s <= 600) {

\_size = "Medium 800";

} else {

\_size = "Large";

}

return \_size;

}

},

instagram: {

get: function(mid, thumb) {

if (thumb) {

return "http://instagr.am/p/" + mid + "/media/?size=t";

} else {

return "http://instagr.am/p/" + mid + "/media/?size=" + VMM.ExternalAPI.instagram.sizes(VMM.master\_config.sizes.api.height);

}

},

sizes: function(s) {

var \_size = "";

if (s <= 150) {

\_size = "t";

} else if (s <= 306) {

\_size = "m";

} else {

\_size = "l";

}

return \_size;

}

},

soundcloud: {

get: function(mid, id) {

var sound = {mid: mid, id: id};

VMM.master\_config.soundcloud.que.push(sound);

VMM.master\_config.soundcloud.active = true;

},

create: function(sound, callback) {

var the\_url = "http://soundcloud.com/oembed?url=" + sound.mid + "&format=js&callback=?";

VMM.getJSON(the\_url, function(d) {

VMM.attachElement("#"+sound.id, d.html);

callback();

});

},

pushQue: function() {

if (VMM.master\_config.soundcloud.que.length > 0) {

VMM.ExternalAPI.soundcloud.create(VMM.master\_config.soundcloud.que[0], VMM.ExternalAPI.soundcloud.pushQue);

VMM.master\_config.soundcloud.que.remove(0);

}

}

},

wikipedia: {

get: function(url, id, lang) {

var api\_obj = {url: url, id: id, lang: lang};

VMM.master\_config.wikipedia.que.push(api\_obj);

VMM.master\_config.wikipedia.active = true;

},

create: function(api\_obj, callback) {

var the\_url = "http://" + api\_obj.lang + ".wikipedia.org/w/api.php?action=query&prop=extracts&redirects=&titles=" + api\_obj.url + "&exintro=1&format=json&callback=?";

callback\_timeout= setTimeout(callback, VMM.master\_config.timers.api, api\_obj);

if ( VMM.Browser.browser == "Explorer" && parseInt(VMM.Browser.version, 10) >= 7 && window.XDomainRequest) {

var temp\_text = "<h4><a href='http://" + VMM.master\_config.language.api.wikipedia + ".wikipedia.org/wiki/" + api\_obj.url + "' target='\_blank'>" + api\_obj.url + "</a></h4>";

temp\_text += "<span class='wiki-source'>" + VMM.master\_config.language.messages.wikipedia + "</span>";

temp\_text += "<p>Wikipedia entry unable to load using Internet Explorer 8 or below.</p>";

VMM.attachElement("#"+api\_obj.id, temp\_text );

}

VMM.getJSON(the\_url, function(d) {

if (d.query) {

var wiki\_extract,

wiki\_title,

\_wiki = "",

wiki\_text = "",

wiki\_number\_of\_paragraphs = 1,

wiki\_text\_array = [];

wiki\_extract = VMM.Util.getObjectAttributeByIndex(d.query.pages, 0).extract;

wiki\_title = VMM.Util.getObjectAttributeByIndex(d.query.pages, 0).title;

if (wiki\_extract.match("<p>")) {

wiki\_text\_array = wiki\_extract.split("<p>");

} else {

wiki\_text\_array.push(wiki\_extract);

}

for(var i = 0; i < wiki\_text\_array.length; i++) {

if (i+1 <= wiki\_number\_of\_paragraphs && i+1 < wiki\_text\_array.length) {

wiki\_text += "<p>" + wiki\_text\_array[i+1];

}

}

\_wiki = "<h4><a href='http://" + VMM.master\_config.language.api.wikipedia + ".wikipedia.org/wiki/" + wiki\_title + "' target='\_blank'>" + wiki\_title + "</a></h4>";

\_wiki += "<span class='wiki-source'>" + VMM.master\_config.language.messages.wikipedia + "</span>";

\_wiki += VMM.Util.linkify\_wikipedia(wiki\_text);

if (wiki\_extract.match("REDIRECT")) {

} else {

VMM.attachElement("#"+api\_obj.id, \_wiki );

}

}

//callback();

})

.error(function(jqXHR, textStatus, errorThrown) {

trace("WIKIPEDIA error");

trace("WIKIPEDIA ERROR: " + textStatus + " " + jqXHR.responseText);

trace(errorThrown);

VMM.attachElement("#"+api\_obj.id, VMM.MediaElement.loadingmessage("<p>Wikipedia is not responding</p>"));

// TRY AGAIN?

clearTimeout(callback\_timeout);

if (VMM.master\_config.wikipedia.tries < 4) {

trace("WIKIPEDIA ATTEMPT " + VMM.master\_config.wikipedia.tries);

trace(api\_obj);

VMM.master\_config.wikipedia.tries++;

VMM.ExternalAPI.wikipedia.create(api\_obj, callback);

} else {

callback();

}

})

.success(function(d) {

VMM.master\_config.wikipedia.tries = 0;

clearTimeout(callback\_timeout);

callback();

});

},

pushQue: function() {

if (VMM.master\_config.wikipedia.que.length > 0) {

trace("WIKIPEDIA PUSH QUE " + VMM.master\_config.wikipedia.que.length);

VMM.ExternalAPI.wikipedia.create(VMM.master\_config.wikipedia.que[0], VMM.ExternalAPI.wikipedia.pushQue);

VMM.master\_config.wikipedia.que.remove(0);

}

}

},

youtube: {

get: function(mid, id, start) {

var the\_url = "http://gdata.youtube.com/feeds/api/videos/" + mid + "?v=2&alt=jsonc&callback=?",

vid = {mid: mid, id: id, start: start};

VMM.master\_config.youtube.que.push(vid);

if (!VMM.master\_config.youtube.active) {

if (!VMM.master\_config.youtube.api\_loaded) {

VMM.LoadLib.js('http://www.youtube.com/player\_api', function() {

trace("YouTube API Library Loaded");

});

}

}

// THUMBNAIL

VMM.getJSON(the\_url, function(d) {

VMM.ExternalAPI.youtube.createThumb(d, vid)

});

},

create: function(vid) {

if (typeof(vid.start) != 'undefined') {

var vidstart = vid.start.toString(),

vid\_start\_minutes = 0,

vid\_start\_seconds = 0;

if (vidstart.match('m')) {

vidstart = vidstart.split("=")[1];

vid\_start\_minutes = parseInt(vidstart.split("m")[0], 10);

vid\_start\_seconds = parseInt(vidstart.split("m")[1].split("s")[0], 10);

vid.start = (vid\_start\_minutes \* 60) + vid\_start\_seconds;

} else {

vid.start = 0;

}

} else {

vid.start = 0;

}

var p = {

active: false,

player: {},

name: vid.id,

playing: false

};

p.player[vid.id] = new YT.Player(vid.id, {

height: '390',

width: '640',

playerVars: {

enablejsapi: 1,

color: 'white',

showinfo: 0,

theme: 'light',

start: vid.start,

rel: 0

},

videoId: vid.mid,

events: {

'onReady': VMM.ExternalAPI.youtube.onPlayerReady,

'onStateChange': VMM.ExternalAPI.youtube.onStateChange

}

});

VMM.master\_config.youtube.array.push(p);

},

createThumb: function(d, vid) {

trace("CREATE THUMB");

trace(d);

trace(vid);

if (typeof d.data != 'undefined') {

var thumb\_id = "#" + vid.id + "\_thumb";

VMM.attachElement(thumb\_id, "<img src='" + d.data.thumbnail.sqDefault + "'>");

}

},

pushQue: function() {

for(var i = 0; i < VMM.master\_config.youtube.que.length; i++) {

VMM.ExternalAPI.youtube.create(VMM.master\_config.youtube.que[i]);

}

VMM.master\_config.youtube.que = [];

},

onAPIReady: function() {

VMM.master\_config.youtube.active = true;

VMM.ExternalAPI.youtube.pushQue();

},

stopPlayers: function() {

for(var i = 0; i < VMM.master\_config.youtube.array.length; i++) {

if (VMM.master\_config.youtube.array[i].playing) {

var the\_name = VMM.master\_config.youtube.array[i].name;

VMM.master\_config.youtube.array[i].player[the\_name].stopVideo();

}

}

},

onStateChange: function(e) {

for(var i = 0; i < VMM.master\_config.youtube.array.length; i++) {

var the\_name = VMM.master\_config.youtube.array[i].name;

if (VMM.master\_config.youtube.array[i].player[the\_name] == e.target) {

if (e.data == YT.PlayerState.PLAYING) {

VMM.master\_config.youtube.array[i].playing = true;

}

}

}

},

onPlayerReady: function(e) {

}

},

vimeo: {

get: function(mid, id) {

var vid = {mid: mid, id: id};

VMM.master\_config.vimeo.que.push(vid);

VMM.master\_config.vimeo.active = true;

},

create: function(vid, callback) {

trace("VIMEO CREATE");

// THUMBNAIL

var the\_url = "http://vimeo.com/api/v2/video/" + vid.mid + ".json";

VMM.getJSON(the\_url, function(d) {

VMM.ExternalAPI.vimeo.createThumb(d, vid);

callback();

});

},

createThumb: function(d, vid) {

trace("VIMEO CREATE THUMB");

var thumb\_id = "#" + vid.id + "\_thumb";

VMM.attachElement(thumb\_id, "<img src='" + d[0].thumbnail\_small + "'>");

},

pushQue: function() {

if (VMM.master\_config.vimeo.que.length > 0) {

VMM.ExternalAPI.vimeo.create(VMM.master\_config.vimeo.que[0], VMM.ExternalAPI.vimeo.pushQue);

VMM.master\_config.vimeo.que.remove(0);

}

/\*

for(var i = 0; i < VMM.master\_config.vimeo.que.length; i++) {

VMM.ExternalAPI.vimeo.create(VMM.master\_config.vimeo.que[i]);

}

VMM.master\_config.vimeo.que = [];

\*/

}

}

}

}

/\* YOUTUBE API READY

Can't find a way to customize this callback and keep it in the VMM namespace

Youtube wants it to be this function.

================================================== \*/

function onYouTubePlayerAPIReady() {

trace("GLOBAL YOUTUBE API CALLED")

VMM.ExternalAPI.youtube.onAPIReady();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.MediaElement.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* MediaElement

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.MediaElement == 'undefined') {

VMM.MediaElement = ({

init: function() {

return this;

},

loadingmessage: function(m) {

return "<div class='loading'><div class='loading-container'><div class='loading-icon'></div>" + "<div class='message'><p>" + m + "</p></div></div></div>";

},

thumbnail: function(data, w, h, uid) {

var \_w = 16,

\_h = 24,

\_uid = "";

if (w != null && w != "") {\_w = w};

if (h != null && h != "") {\_h = h};

if (uid != null && uid != "") {\_uid = uid};

if (data.media != null && data.media != "") {

var \_valid = true,

mediaElem = "",

m = VMM.MediaType(data.media); //returns an object with .type and .id

// DETERMINE THUMBNAIL OR ICON

if (data.thumbnail != null && data.thumbnail != "") {

trace("CUSTOM THUMB");

mediaElem = "<div class='thumbnail thumb-custom' id='" + uid + "\_custom\_thumb'><img src='" + data.thumbnail + "'></div>";

return mediaElem;

} else if (m.type == "image") {

mediaElem = "<div class='thumbnail thumb-photo'></div>";

return mediaElem;

} else if (m.type == "flickr") {

mediaElem = "<div class='thumbnail thumb-photo' id='" + uid + "\_thumb'></div>";

return mediaElem;

} else if (m.type == "instagram") {

mediaElem = "<div class='thumbnail thumb-instagram' id='" + uid + "\_thumb'><img src='" + VMM.ExternalAPI.instagram.get(m.id, true) + "'></div>";

return mediaElem;

} else if (m.type == "youtube") {

mediaElem = "<div class='thumbnail thumb-youtube' id='" + uid + "\_thumb'></div>";

return mediaElem;

} else if (m.type == "googledoc") {

mediaElem = "<div class='thumbnail thumb-document'></div>";

return mediaElem;

} else if (m.type == "vimeo") {

mediaElem = "<div class='thumbnail thumb-vimeo' id='" + uid + "\_thumb'></div>";

return mediaElem;

} else if (m.type == "dailymotion") {

mediaElem = "<div class='thumbnail thumb-video'></div>";

return mediaElem;

} else if (m.type == "twitter"){

mediaElem = "<div class='thumbnail thumb-twitter'></div>";

return mediaElem;

} else if (m.type == "twitter-ready") {

mediaElem = "<div class='thumbnail thumb-twitter'></div>";

return mediaElem;

} else if (m.type == "soundcloud") {

mediaElem = "<div class='thumbnail thumb-audio'></div>";

return mediaElem;

} else if (m.type == "google-map") {

mediaElem = "<div class='thumbnail thumb-map'></div>";

return mediaElem;

} else if (m.type == "googleplus") {

mediaElem = "<div class='thumbnail thumb-googleplus'></div>";

return mediaElem;

} else if (m.type == "wikipedia") {

mediaElem = "<div class='thumbnail thumb-wikipedia'></div>";

return mediaElem;

} else if (m.type == "storify") {

mediaElem = "<div class='thumbnail thumb-storify'></div>";

return mediaElem;

} else if (m.type == "quote") {

mediaElem = "<div class='thumbnail thumb-quote'></div>";

return mediaElem;

} else if (m.type == "unknown") {

if (m.id.match("blockquote")) {

mediaElem = "<div class='thumbnail thumb-quote'></div>";

} else {

mediaElem = "<div class='thumbnail thumb-plaintext'></div>";

}

return mediaElem;

} else if (m.type == "website") {

mediaElem = "<div class='thumbnail thumb-website'></div>";

return mediaElem;

} else {

mediaElem = "<div class='thumbnail thumb-plaintext'></div>";

return mediaElem;

}

}

},

create: function(data, uid) {

var \_valid = false,

//loading\_messege = "<span class='messege'><p>" + VMM.master\_config.language.messages.loading + "</p></span>";

loading\_messege = VMM.MediaElement.loadingmessage(VMM.master\_config.language.messages.loading + "...");

if (data.media != null && data.media != "") {

var mediaElem = "", captionElem = "", creditElem = "", \_id = "", isTextMedia = false, m;

m = VMM.MediaType(data.media); //returns an object with .type and .id

\_valid = true;

// CREDIT

if (data.credit != null && data.credit != "") {

creditElem = "<div class='credit'>" + VMM.Util.linkify\_with\_twitter(data.credit, "\_blank") + "</div>";

}

// CAPTION

if (data.caption != null && data.caption != "") {

captionElem = "<div class='caption'>" + VMM.Util.linkify\_with\_twitter(data.caption, "\_blank") + "</div>";

}

// IMAGE

if (m.type == "image") {

mediaElem = "<div class='media-image media-shadow'><img src='" + m.id + "' class='media-image'></div>";

// FLICKR

} else if (m.type == "flickr") {

//mediaElem = "<div class='media-image media-shadow' id='" + uid + "'>" + loading\_messege + "</div>";

mediaElem = "<div class='media-image media-shadow'><a href='" + m.link + "' target='\_blank'><img id='" + uid + "'></a></div>";

VMM.ExternalAPI.flickr.get(m.id, uid, m.link);

// INSTAGRAM

} else if (m.type == "instagram") {

mediaElem = "<div class='media-image media-shadow'><a href='" + m.link + "' target='\_blank'><img src='" + VMM.ExternalAPI.instagram.get(m.id) + "'></a></div>";

//VMM.ExternalAPI.instagram.get(m.id, uid);

// GOOGLE DOCS

} else if (m.type == "googledoc") {

mediaElem = "<div class='media-frame media-shadow doc' id='" + uid + "'>" + loading\_messege + "</div>";

VMM.ExternalAPI.googledocs.get(m.id, uid);

// YOUTUBE

} else if (m.type == "youtube") {

mediaElem = "<div class='media-shadow'><div class='media-frame video youtube' id='" + uid + "'>" + loading\_messege + "</div></div>";

VMM.ExternalAPI.youtube.get(m.id, uid, m.start);

// VIMEO

} else if (m.type == "vimeo") {

mediaElem = "<div class='media-shadow'><iframe class='media-frame video vimeo' autostart='false' frameborder='0' width='100%' height='100%' src='http://player.vimeo.com/video/" + m.id + "?title=0&amp;byline=0&amp;portrait=0&amp;color=ffffff'></iframe></div>";

VMM.ExternalAPI.vimeo.get(m.id, uid);

// DAILYMOTION

} else if (m.type == "dailymotion") {

mediaElem = "<div class='media-shadow'><iframe class='media-frame video dailymotion' autostart='false' frameborder='0' width='100%' height='100%' src='http://www.dailymotion.com/embed/video/" + m.id + "'></iframe></div>";

// TWITTER

} else if (m.type == "twitter"){

mediaElem = "<div class='twitter' id='" + uid + "'>" + loading\_messege + "</div>";

isTextMedia = true;

VMM.ExternalAPI.twitter.get(m.id, uid);

// TWITTER

} else if (m.type == "twitter-ready") {

isTextMedia = true;

mediaElem = m.id;

// SOUNDCLOUD

} else if (m.type == "soundcloud") {

mediaElem = "<div class='media-frame media-shadow soundcloud' id='" + uid + "'>" + loading\_messege + "</div>";

VMM.ExternalAPI.soundcloud.get(m.id, uid);

// GOOGLE MAPS

} else if (m.type == "google-map") {

mediaElem = "<div class='media-frame media-shadow map' id='" + uid + "'>" + loading\_messege + "</div>";

VMM.ExternalAPI.googlemaps.get(m.id, uid);

// GOOGLE PLUS

} else if (m.type == "googleplus") {

\_id = "googleplus\_" + m.id;

mediaElem = "<div class='googleplus' id='" + \_id + "'>" + loading\_messege + "</div>";

isTextMedia = true;

VMM.ExternalAPI.googleplus.get(m.user, m.id, uid);

// WIKIPEDIA

} else if (m.type == "wikipedia") {

mediaElem = "<div class='wikipedia' id='" + uid + "'>" + loading\_messege + "</div>";

isTextMedia = true;

VMM.ExternalAPI.wikipedia.get(m.id, uid, m.lang);

// STORIFY

} else if (m.type == "storify") {

isTextMedia = true;

mediaElem = "<div class='plain-text-quote'>" + m.id + "</div>";

// QUOTE

} else if (m.type == "quote") {

isTextMedia = true;

mediaElem = "<div class='plain-text-quote'>" + m.id + "</div>";

// UNKNOWN

} else if (m.type == "unknown") {

trace("NO KNOWN MEDIA TYPE FOUND TRYING TO JUST PLACE THE HTML");

isTextMedia = true;

mediaElem = "<div class='plain-text'><div class='container'>" + VMM.Util.properQuotes(m.id) + "</div></div>";

// WEBSITE

} else if (m.type == "website") {

//mediaElem = "<div class='media-shadow'><iframe class='media-frame website' frameborder='0' autostart='false' width='100%' height='100%' scrolling='yes' marginheight='0' marginwidth='0' src='" + m.id + "'></iframe></div>";

//mediaElem = "<a href='" + m.id + "' target='\_blank'>" + "<img src='http://api.snapito.com/free/lc?url=" + m.id + "'></a>";

mediaElem = "<div class='media-shadow website'><a href='" + m.id + "' target='\_blank'>" + "<img src='http://api1.thumbalizr.com/?url=" + m.id.replace(/[\./]$/g, "") + "&width=300' class='media-image'></a></div>";

// NO MATCH

} else {

trace("NO KNOWN MEDIA TYPE FOUND");

trace(m.type);

}

// WRAP THE MEDIA ELEMENT

mediaElem = "<div class='media-container' >" + mediaElem + creditElem + captionElem + "</div>";

// RETURN

if (isTextMedia) {

return "<div class='text-media'><div class='media-wrapper'>" + mediaElem + "</div></div>";

} else {

return "<div class='media-wrapper'>" + mediaElem + "</div>";

}

};

}

}).init();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.MediaType.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* MediaType

Determines the type of media the url string is.

returns an object with .type and .id

the id is a key piece of information needed to make

the request of the api.

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.MediaType == 'undefined') {

VMM.MediaType = function(d) {

var success = false,

media = {

type: "unknown",

id: "",

start: 0,

link: "",

lang: "",

uniqueid: VMM.Util.unique\_ID(6)

};

if (d.match("div class='twitter'")) {

media.type = "twitter-ready";

media.id = d;

success = true;

} else if (d.match('(www.)?youtube|youtu\.be')) {

if (d.match('v=')) {

media.id = VMM.Util.getUrlVars(d)["v"];

media.start = VMM.Util.getUrlVars(d)["t"];

} else if (d.match('\/embed\/')) {

media.id = d.split("embed\/")[1].split(/[?&]/)[0];

media.start = d.split("embed\/")[1].split(/[?&]/)[1];

} else {

media.id = d.split(/v\/|v=|youtu\.be\//)[1].split(/[?&]/)[0];

media.start = d.split(/v\/|v=|youtu\.be\//)[1].split(/[?&]/)[1];

}

media.type = "youtube";

success = true;

} else if (d.match('(player.)?vimeo\.com')) {

media.type = "vimeo";

media.id = d.split(/video\/|\/\/vimeo\.com\//)[1].split(/[?&]/)[0];;

success = true;

} else if (d.match('(www.)?dailymotion\.com')) {

media.id = d.split(/video\/|\/\/dailymotion\.com\//)[1];

media.type = "dailymotion";

success = true;

} else if (d.match('(player.)?soundcloud\.com')) {

media.type = "soundcloud";

media.id = d;

success = true;

} else if (d.match('(www.)?twitter\.com') && d.match('status') ) {

if (d.match("status\/")) {

media.id = d.split("status\/")[1];

} else if (d.match("statuses\/")) {

media.id = d.split("statuses\/")[1];

} else {

media.id = "";

}

media.type = "twitter";

success = true;

} else if (d.match("maps.google") && !d.match("staticmap")) {

media.type = "google-map";

media.id = d.split(/src=['|"][^'|"]\*?['|"]/gi);

success = true;

} else if (d.match("plus.google")) {

media.type = "googleplus";

media.id = d.split("/posts/")[1];

//https://plus.google.com/u/0/112374836634096795698/posts/bRJSvCb5mUU

//https://plus.google.com/107096716333816995401/posts/J5iMpEDHWNL

if (d.split("/posts/")[0].match("u/0/")) {

media.user = d.split("u/0/")[1].split("/posts")[0];

} else {

media.user = d.split("google.com/")[1].split("/posts/")[0];

}

success = true;

} else if (d.match("flickr.com/photos")) {

media.type = "flickr";

media.id = d.split("photos\/")[1].split("/")[1];

media.link = d;

success = true;

} else if (d.match("instagr.am/p/")) {

media.type = "instagram";

media.link = d;

media.id = d.split("\/p\/")[1].split("/")[0];

success = true;

} else if (d.match(/jpg|jpeg|png|gif/i) || d.match("staticmap") || d.match("yfrog.com") || d.match("twitpic.com")) {

media.type = "image";

media.id = d;

success = true;

} else if (VMM.FileExtention.googleDocType(d)) {

media.type = "googledoc";

media.id = d;

success = true;

} else if (d.match('(www.)?wikipedia\.org')) {

media.type = "wikipedia";

//media.id = d.split("wiki\/")[1];

var wiki\_id = d.split("wiki\/")[1].split("#")[0].replace("\_", " ");

media.id = wiki\_id.replace(" ", "%20");

media.lang = d.split("//")[1].split(".wikipedia")[0];

success = true;

} else if (d.indexOf('http://') == 0) {

media.type = "website";

media.id = d;

success = true;

} else if (d.match('storify')) {

media.type = "storify";

media.id = d;

success = true;

} else if (d.match('blockquote')) {

media.type = "quote";

media.id = d;

success = true;

} else {

trace("unknown media");

media.type = "unknown";

media.id = d;

success = true;

}

if (success) {

return media;

} else {

trace("No valid media id detected");

trace(d);

}

return false;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Media.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Media

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Media == 'undefined') {

// something = new VMM.Media(parent, w, h, {thedata});

VMM.Media = function(parent, w, h, thedata) {

/\* PRIVATE VARS

================================================== \*/

var data = {}; // HOLDS DATA

var \_valid = false;

var config = {

width: 720,

height: 400,

content\_width: 720,

content\_height: 400,

ease: "easeInOutExpo",

duration: 1000,

spacing: 15

};

/\* ELEMENTS

================================================== \*/

var $media = "";

var $container = "";

var $mediacontainer = "";

var $mediaelement = "";

var layout = parent; // expecting media div

if (w != null && w != "") {config.width = w};

if (h != null && h != "") {config.height = h};

/\*

if (typeof thedata != "undefined") {

data = thedata;

this.init(data);

}

\*/

/\* PUBLIC FUNCTIONS

================================================== \*/

this.init = function(d) {

if(typeof d != 'undefined') {

this.setData(d);

} else {

trace("WAITING ON DATA");

}

};

var build = function(media, caption, credit) {

$media = VMM.appendAndGetElement(layout, "<div>", "media");

$container = VMM.appendAndGetElement($media, "<div>", "container");

$mediacontainer = VMM.appendAndGetElement($container, "<div>", "media-container");

if (data.media != null && data.media != "") {

\_valid = true;

var m = {};

m = VMM.MediaType(data.media); //returns an object with .type and .id

if (m.type == "image") {

VMM.appendElement($mediacontainer, "<img src='" + m.id + "'>");

} else if (m.type == "youtube") {

VMM.appendElement($mediacontainer, "<iframe frameborder='0' src='http://www.youtube.com/embed/" + m.id + "?&rel=0&theme=light&showinfo=0&hd=1&autohide=0&color=white' allowfullscreen>");

} else if (m.type == "vimeo") {

VMM.appendElement($mediacontainer, "<iframe frameborder='0' src='http://player.vimeo.com/video/" + m.id + "?title=0&amp;byline=0&amp;portrait=0&amp;color=ffffff'>");

} else {

}

// CREDIT

if (data.credit != null && data.credit != "") {

VMM.appendElement($container, VMM.createElement("div", data.credit, "credit"));

}

// CAPTION

if (data.caption != null && data.caption != "") {

VMM.appendElement($container, VMM.createElement("div", data.caption, "caption"));

}

}

};

/\* GETTERS AND SETTERS

================================================== \*/

this.setData = function(d) {

if(typeof d != 'undefined') {

data = d;

build();

} else{

trace("NO DATA");

}

};

/\* RESIZE

================================================== \*/

function reSize() {

}

}

// Less expensive to use prototype

VMM.Media.prototype.height = function(h) {

if (h != null && h != "") {

config.height = h;

reSize();

} else {

return config.height;

}

};

VMM.Media.prototype.width = function(w) {

if (w != null && w != "") {

config.width = w;

reSize();

} else {

return config.width;

}

};

/\* GETTERS AND SETTERS

================================================== \*/

VMM.Media.prototype.getData = function() {

return data;

};

VMM.Media.prototype.setConfig = function(d) {

if(typeof d != 'undefined') {

config = d;

} else{

trace("NO CONFIG DATA");

}

};

VMM.Media.prototype.getConfig = function() {

return config;

};

VMM.Media.prototype.setSize = function(w, h) {

if (w != null) {config.width = w};

if (h != null) {config.height = h};

if (\_active) {

reSize();

}

}

VMM.Media.prototype.active = function() {

return \_active;

};

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.TextElement.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* TextElement

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.TextElement == 'undefined') {

VMM.TextElement = ({

init: function() {

return this;

},

create: function(data) {

return data;

}

}).init();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.DragSlider.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* DRAG SLIDER

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.DragSlider == 'undefined') {

// VMM.DragSlider.createSlidePanel(drag\_object, move\_object, w, padding, sticky);

// VMM.DragSlider.cancelSlide();

VMM.DragSlider = function() {

var drag = {

element: "",

element\_move: "",

constraint: "",

sliding: false,

pagex: {

start: 0,

end: 0

},

left: {

start: 0,

end: 0

},

time: {

start: 0,

end: 0

},

touch: false,

ease: "easeOutExpo"

},

dragevent = {

down: "mousedown",

up: "mouseup",

leave: "mouseleave",

move: "mousemove"

},

mousedrag = {

down: "mousedown",

up: "mouseup",

leave: "mouseleave",

move: "mousemove"

},

touchdrag = {

down: "touchstart",

up: "touchend",

leave: "mouseleave",

move: "touchmove"

};

this.createPanel = function(drag\_object, move\_object, constraint, touch) {

drag.element = drag\_object;

drag.element\_move = move\_object;

if ( constraint != null && constraint != "") {

drag.constraint = constraint;

} else {

drag.constraint = false;

}

if ( touch) {

drag.touch = touch;

} else {

drag.touch = false;

}

trace("TOUCH" + drag.touch);

if (drag.touch) {

dragevent = touchdrag;

} else {

dragevent = mousedrag;

}

makeDraggable(drag.element, drag.element\_move);

}

this.updateConstraint = function(constraint) {

trace("updateConstraint");

drag.constraint = constraint;

}

var makeDraggable = function(drag\_object, move\_object) {

VMM.bindEvent(drag\_object, onDragStart, dragevent.down, {element: move\_object, delement: drag\_object});

VMM.bindEvent(drag\_object, onDragEnd, dragevent.up, {element: move\_object, delement: drag\_object});

VMM.bindEvent(drag\_object, onDragLeave, dragevent.leave, {element: move\_object, delement: drag\_object});

}

this.cancelSlide = function(e) {

VMM.unbindEvent(drag.element, onDragMove, dragevent.move);

return true;

}

var onDragLeave = function(e) {

VMM.unbindEvent(e.data.delement, onDragMove, dragevent.move);

if (!drag.touch) {

e.preventDefault();

}

e.stopPropagation();

if (drag.sliding) {

drag.sliding = false;

dragEnd(e.data.element, e.data.delement, e);

return false;

} else {

return true;

}

}

var onDragStart = function(e) {

dragStart(e.data.element, e.data.delement, e);

if (!drag.touch) {

e.preventDefault();

}

e.stopPropagation();

return true;

}

var onDragEnd = function(e) {

if (!drag.touch) {

e.preventDefault();

}

e.stopPropagation();

if (drag.sliding) {

drag.sliding = false;

dragEnd(e.data.element, e.data.delement, e);

return false;

} else {

return true;

}

}

var onDragMove = function(e) {

dragMove(e.data.element, e);

e.preventDefault();

e.stopPropagation();

return false;

}

var dragStart = function(elem, delem, e) {

if (drag.touch) {

trace("IS TOUCH")

VMM.Lib.css(elem, '-webkit-transition-duration', '0');

drag.pagex.start = e.originalEvent.touches[0].screenX;

} else {

drag.pagex.start = e.pageX;

}

drag.left.start = getLeft(elem);

drag.time.start = new Date().getTime();

VMM.Lib.stop(elem);

VMM.bindEvent(delem, onDragMove, dragevent.move, {element: elem});

}

var dragEnd = function(elem, delem, e) {

VMM.unbindEvent(delem, onDragMove, dragevent.move);

dragMomentum(elem, e);

}

var dragMove = function(elem, e) {

drag.sliding = true;

if (drag.touch) {

drag.pagex.end = e.originalEvent.touches[0].screenX;

} else {

drag.pagex.end = e.pageX;

}

drag.left.end = getLeft(elem);

VMM.Lib.css(elem, 'left', -(drag.pagex.start - drag.pagex.end - drag.left.start));

}

var dragMomentum = function(elem, e) {

var drag\_info = {

left: drag.left.end,

left\_adjust: 0,

change: {

x: 0

},

time: (new Date().getTime() - drag.time.start) \* 10,

time\_adjust: (new Date().getTime() - drag.time.start) \* 10

},

multiplier = 3000;

if (drag.touch) {

multiplier = 6000;

}

drag\_info.change.x = multiplier \* (Math.abs(drag.pagex.end) - Math.abs(drag.pagex.start));

drag\_info.left\_adjust = Math.round(drag\_info.change.x / drag\_info.time);

drag\_info.left = Math.min(drag\_info.left + drag\_info.left\_adjust);

if (drag.constraint) {

if (drag\_info.left > drag.constraint.left) {

drag\_info.left = drag.constraint.left;

if (drag\_info.time > 5000) {

drag\_info.time = 5000;

}

} else if (drag\_info.left < drag.constraint.right) {

drag\_info.left = drag.constraint.right;

if (drag\_info.time > 5000) {

drag\_info.time = 5000;

}

}

}

VMM.fireEvent(elem, "DRAGUPDATE", [drag\_info]);

if (drag\_info.time > 0) {

if (drag.touch) {

//VMM.Lib.css(elem, '-webkit-transition-property', 'left');

//VMM.Lib.css(elem, '-webkit-transition-duration', drag\_info.time);

//VMM.Lib.css(elem, 'left', drag\_info.left);

//VMM.Lib.animate(elem, drag\_info.time, "easeOutQuad", {"left": drag\_info.left});

VMM.Lib.animate(elem, drag\_info.time, "easeOutCirc", {"left": drag\_info.left});

//VMM.Lib.css(elem, 'webkitTransition', '');

//VMM.Lib.css(elem, 'webkitTransition', '-webkit-transform ' + drag\_info.time + 'ms cubic-bezier(0.33, 0.66, 0.66, 1)');

//VMM.Lib.css(elem, 'webkitTransform', 'translate3d(' + drag\_info.left + 'px, 0, 0)');

} else {

VMM.Lib.animate(elem, drag\_info.time, drag.ease, {"left": drag\_info.left});

}

}

}

var getLeft = function(elem) {

return parseInt(VMM.Lib.css(elem, 'left').substring(0, VMM.Lib.css(elem, 'left').length - 2), 10);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Slider.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Slider

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Slider == 'undefined') {

VMM.Slider = function(parent, parent\_config) {

var events = {}, config;

var $slider, $slider\_mask, $slider\_container, $slides\_items;

var data = [], slides = [], slide\_positions = [];

var slides\_content = "";

var current\_slide = 0;

var current\_width = 960;

var touch = {move: false, x: 10, y:0, off: 0, dampen: 48};

var content = "";

var \_active = false;

var layout = parent;

var navigation = {nextBtn:"", prevBtn:"", nextDate:"", prevDate:"", nextTitle:"", prevTitle:""};

var timer;

// CONFIG

if(typeof VMM.Timeline != 'undefined') {

config = VMM.Timeline.Config;

} else {

config = {

preload: 4,

current\_slide: 0,

interval: 10,

something: 0,

width: 720,

height: 400,

ease: "easeInOutExpo",

duration: 1000,

timeline: false,

spacing: 15,

slider: {

width: 720,

height: 400,

content: {

width: 720,

height: 400,

padding: 130

},

nav: {

width: 100,

height: 200

}

}

};

}

/\* PUBLIC VARS

================================================== \*/

this.ver = "0.6";

config.slider.width = config.width;

config.slider.height = config.height;

/\* PUBLIC FUNCTIONS

================================================== \*/

this.init = function(d) {

slides = [];

slide\_positions = [];

if(typeof d != 'undefined') {

this.setData(d);

} else {

trace("WAITING ON DATA");

}

};

this.width = function(w) {

if (w != null && w != "") {

config.slider.width = w;

reSize();

} else {

return config.slider.width;

}

}

this.height = function(h) {

if (h != null && h != "") {

config.slider.height = h;

reSize();

} else {

return config.slider.height;

}

}

/\* GETTERS AND SETTERS

================================================== \*/

this.setData = function(d) {

if(typeof d != 'undefined') {

data = d;

build();

} else{

trace("NO DATA");

}

};

this.getData = function() {

return data;

};

this.setConfig = function(d) {

if(typeof d != 'undefined') {

config = d;

} else{

trace("NO CONFIG DATA");

}

}

this.getConfig = function() {

return config;

};

this.setSize = function(w, h) {

if (w != null) {config.slider.width = w};

if (h != null) {config.slider.height = h};

if (\_active) {

reSize();

}

}

this.active = function() {

return \_active;

};

this.getCurrentNumber = function() {

return current\_slide;

};

this.setSlide = function(n) {

goToSlide(n);

};

/\* ON EVENT

================================================== \*/

function onConfigSet() {

trace("onConfigSet");

};

function reSize(go\_to\_slide, from\_start) {

var \_go\_to\_slide = true;

var \_from\_start = false;

if (go\_to\_slide != null) {\_go\_to\_slide = go\_to\_slide};

if (from\_start != null) {\_from\_start = from\_start};

current\_width = config.slider.width;

config.slider.nav.height = VMM.Lib.height(navigation.prevBtnContainer);

config.slider.content.width = current\_width - (config.slider.content.padding \*2);

VMM.Lib.width($slides\_items, (slides.length \* config.slider.content.width));

if (\_from\_start) {

var \_pos = slides[current\_slide].leftpos();

VMM.Lib.css($slider\_container, "left", \_pos);

}

// RESIZE SLIDES

sizeSlides();

// POSITION SLIDES

positionSlides();

// POSITION NAV

VMM.Lib.css(navigation.nextBtn, "left", (current\_width - config.slider.nav.width));

VMM.Lib.height(navigation.prevBtn, config.slider.height);

VMM.Lib.height(navigation.nextBtn, config.slider.height);

VMM.Lib.css(navigation.nextBtnContainer, "top", ( (config.slider.height/2) - (config.slider.nav.height/2) ) + 10 );

VMM.Lib.css(navigation.prevBtnContainer, "top", ( (config.slider.height/2) - (config.slider.nav.height/2) ) + 10 );

// Animate Changes

VMM.Lib.height($slider\_mask, config.slider.height);

VMM.Lib.width($slider\_mask, current\_width);

if (\_go\_to\_slide) {

goToSlide(current\_slide, "linear", 1);

};

if (current\_slide == 0) {

VMM.Lib.visible(navigation.prevBtn, false);

}

}

/\* NAVIGATION

================================================== \*/

function onNextClick(e) {

if (current\_slide == slides.length - 1) {

VMM.Lib.animate($slider\_container, config.duration, config.ease, {"left": -(slides[current\_slide].leftpos()) } );

} else {

goToSlide(current\_slide+1);

upDate();

}

}

function onPrevClick(e) {

if (current\_slide == 0) {

goToSlide(current\_slide);

} else {

goToSlide(current\_slide-1);

upDate();

}

}

function onKeypressNav(e) {

switch(e.keyCode) {

case 39:

// RIGHT ARROW

onNextClick(e);

break;

case 37:

// LEFT ARROW

onPrevClick(e);

break;

}

}

function onTouchUpdate(e, b) {

if (slide\_positions.length == 0) {

for(var i = 0; i < slides.length; i++) {

slide\_positions.push( slides[i].leftpos() );

}

}

if (typeof b.left == "number") {

var \_pos = b.left;

var \_slide\_pos = -(slides[current\_slide].leftpos());

if (\_pos < \_slide\_pos - (config.slider\_width/3)) {

onNextClick();

} else if (\_pos > \_slide\_pos + (config.slider\_width/3)) {

onPrevClick();

} else {

VMM.Lib.animate($slider\_container, config.duration, config.ease, {"left": \_slide\_pos });

}

} else {

VMM.Lib.animate($slider\_container, config.duration, config.ease, {"left": \_slide\_pos });

}

if (typeof b.top == "number") {

VMM.Lib.animate($slider\_container, config.duration, config.ease, {"top": -b.top});

} else {

}

};

/\* UPDATE

================================================== \*/

function upDate() {

config.current\_slide = current\_slide;

VMM.fireEvent(layout, "UPDATE");

};

/\* GET DATA

================================================== \*/

var getData = function(d) {

data = d;

};

/\* BUILD SLIDES

================================================== \*/

var buildSlides = function(d) {

VMM.attachElement($slides\_items, "");

slides = [];

for(var i = 0; i < d.length; i++) {

var \_slide = new VMM.Slider.Slide(d[i], $slides\_items);

//\_slide.show();

slides.push(\_slide);

}

}

var preloadSlides = function(skip) {

if (skip) {

preloadTimeOutSlides();

} else {

for(var k = 0; k < slides.length; k++) {

slides[k].clearTimers();

}

timer = setTimeout(preloadTimeOutSlides, config.duration);

}

}

var preloadTimeOutSlides = function() {

for(var k = 0; k < slides.length; k++) {

slides[k].enqueue = true;

}

for(var j = 0; j < config.preload; j++) {

if ( !((current\_slide + j) > slides.length - 1)) {

slides[current\_slide + j].show();

slides[current\_slide + j].enqueue = false;

}

if ( !( (current\_slide - j) < 0 ) ) {

slides[current\_slide - j].show();

slides[current\_slide - j].enqueue = false;

}

}

if (slides.length > 50) {

for(var i = 0; i < slides.length; i++) {

if (slides[i].enqueue) {

slides[i].hide();

}

}

}

sizeSlides();

}

var sizeSlide = function(slide\_id) {

}

/\* SIZE SLIDES

================================================== \*/

var sizeSlides = function() {

var layout\_text\_media = ".slider-item .layout-text-media .media .media-container ",

layout\_media = ".slider-item .layout-media .media .media-container ",

layout\_both = ".slider-item .media .media-container",

layout\_caption = ".slider-item .media .media-container .media-shadow .caption",

mediasize = {

text\_media: {

width: (config.slider.content.width/100) \* 60,

height: config.slider.height - 60,

video: {

width: 0,

height: 0

},

text: {

width: ((config.slider.content.width/100) \* 40) - 30,

height: config.slider.height

}

},

media: {

width: config.slider.content.width,

height: config.slider.height - 110,

video: {

width: 0,

height: 0

}

}

};

VMM.master\_config.sizes.api.width = mediasize.media.width;

VMM.master\_config.sizes.api.height = mediasize.media.height;

mediasize.text\_media.video = VMM.Util.ratio.fit(mediasize.text\_media.width, mediasize.text\_media.height, 16, 9);

mediasize.media.video = VMM.Util.ratio.fit(mediasize.media.width, mediasize.media.height, 16, 9);

VMM.Lib.css(".slider-item", "width", config.slider.content.width );

VMM.Lib.height(".slider-item", config.slider.height);

// HANDLE SMALLER SIZES

var is\_skinny = false;

if (current\_width <= 640) {

is\_skinny = true;

} else if (VMM.Browser.device == "mobile" && VMM.Browser.orientation == "portrait") {

is\_skinny = true;

} else if (VMM.Browser.device == "tablet" && VMM.Browser.orientation == "portrait") {

//is\_skinny = true;

}

if (is\_skinny) {

mediasize.text\_media.width = config.slider.content.width;

mediasize.text\_media.height = ((config.slider.height/100) \* 50 ) - 50;

mediasize.media.height = ((config.slider.height/100) \* 70 ) - 40;

mediasize.text\_media.video = VMM.Util.ratio.fit(mediasize.text\_media.width, mediasize.text\_media.height, 16, 9);

mediasize.media.video = VMM.Util.ratio.fit(mediasize.media.width, mediasize.media.height, 16, 9);

VMM.Lib.css(".slider-item .layout-text-media .text", "width", "100%" );

VMM.Lib.css(".slider-item .layout-text-media .text", "display", "block" );

VMM.Lib.css(".slider-item .layout-text-media .text .container", "display", "block" );

VMM.Lib.css(".slider-item .layout-text-media .text .container", "width", config.slider.content.width );

VMM.Lib.css(".slider-item .layout-text-media .media", "float", "none" );

VMM.Lib.addClass(".slider-item .content-container", "pad-top");

VMM.Lib.css(".slider-item .media blockquote p", "line-height", "18px" );

VMM.Lib.css(".slider-item .media blockquote p", "font-size", "16px" );

VMM.Lib.css(".slider-item", "overflow-y", "auto" );

} else {

VMM.Lib.css(".slider-item .layout-text-media .text", "width", "40%" );

VMM.Lib.css(".slider-item .layout-text-media .text", "display", "table-cell" );

VMM.Lib.css(".slider-item .layout-text-media .text .container", "display", "table-cell" );

VMM.Lib.css(".slider-item .layout-text-media .text .container", "width", "auto" );

VMM.Lib.css(".slider-item .layout-text-media .text .container .start", "width", mediasize.text\_media.text.width );

//VMM.Lib.addClass(".slider-item .content-container", "pad-left");

VMM.Lib.removeClass(".slider-item .content-container", "pad-top");

VMM.Lib.css(".slider-item .layout-text-media .media", "float", "left" );

VMM.Lib.css(".slider-item .layout-text-media", "display", "table" );

VMM.Lib.css(".slider-item .media blockquote p", "line-height", "36px" );

VMM.Lib.css(".slider-item .media blockquote p", "font-size", "28px" );

VMM.Lib.css(".slider-item", "display", "table" );

VMM.Lib.css(".slider-item", "overflow-y", "auto" );

}

// MEDIA FRAME

VMM.Lib.css( layout\_text\_media + ".media-frame", "max-width", mediasize.text\_media.width);

VMM.Lib.height( layout\_text\_media + ".media-frame", mediasize.text\_media.height);

VMM.Lib.width( layout\_text\_media + ".media-frame", mediasize.text\_media.width);

// WEBSITES

//VMM.Lib.css( layout\_both + ".website", "max-width", 300 );

// IMAGES

VMM.Lib.css( layout\_text\_media + "img", "max-height", mediasize.text\_media.height );

VMM.Lib.css( layout\_media + "img", "max-height", mediasize.media.height );

// FIX FOR NON-WEBKIT BROWSERS

VMM.Lib.css( layout\_text\_media + "img", "max-width", mediasize.text\_media.width );

VMM.Lib.css( layout\_text\_media + ".avatar img", "max-width", 32 );

VMM.Lib.css( layout\_text\_media + ".avatar img", "max-height", 32 );

VMM.Lib.css( layout\_media + ".avatar img", "max-width", 32 );

VMM.Lib.css( layout\_media + ".avatar img", "max-height", 32 );

VMM.Lib.css( layout\_text\_media + ".article-thumb", "max-width", "50%" );

//VMM.Lib.css( layout\_text\_media + ".article-thumb", "max-height", 100 );

VMM.Lib.css( layout\_media + ".article-thumb", "max-width", 200 );

//VMM.Lib.css( layout\_media + ".article-thumb", "max-height", 100 );

// IFRAME FULL SIZE VIDEO

VMM.Lib.width( layout\_text\_media + ".media-frame", mediasize.text\_media.video.width);

VMM.Lib.height( layout\_text\_media + ".media-frame", mediasize.text\_media.video.height);

VMM.Lib.width( layout\_media + ".media-frame", mediasize.media.video.width);

VMM.Lib.height( layout\_media + ".media-frame", mediasize.media.video.height);

VMM.Lib.css( layout\_media + ".media-frame", "max-height", mediasize.media.video.height);

VMM.Lib.css( layout\_media + ".media-frame", "max-width", mediasize.media.video.width);

// SOUNDCLOUD

VMM.Lib.height( layout\_media + ".soundcloud", 168);

VMM.Lib.height( layout\_text\_media + ".soundcloud", 168);

VMM.Lib.width( layout\_media + ".soundcloud", mediasize.media.width);

VMM.Lib.width( layout\_text\_media + ".soundcloud", mediasize.text\_media.width);

VMM.Lib.css( layout\_both + ".soundcloud", "max-height", 168 );

// MAPS

VMM.Lib.height( layout\_text\_media + ".map", mediasize.text\_media.height);

VMM.Lib.css( layout\_media + ".map", "max-height", mediasize.media.height);

VMM.Lib.width( layout\_media + ".map", mediasize.media.width);

// DOCS

VMM.Lib.height( layout\_text\_media + ".doc", mediasize.text\_media.height);

VMM.Lib.height( layout\_media + ".doc", mediasize.media.height);

// IE8 NEEDS THIS

VMM.Lib.width( layout\_media + ".wikipedia", mediasize.media.width);

VMM.Lib.width( layout\_media + ".twitter", mediasize.media.width);

VMM.Lib.width( layout\_media + ".plain-text-quote", mediasize.media.width);

VMM.Lib.width( layout\_media + ".plain-text", mediasize.media.width);

// CAPTION WIDTH

VMM.Lib.css( layout\_text\_media + ".caption", "max-width", mediasize.text\_media.video.width);

VMM.Lib.css( layout\_media + ".caption", "max-width", mediasize.media.video.width);

// MAINTAINS VERTICAL CENTER IF IT CAN

for(var i = 0; i < slides.length; i++) {

slides[i].layout(is\_skinny);

if (slides[i].content\_height() > config.slider.height + 20) {

slides[i].css("display", "block");

} else {

slides[i].css("display", "table");

}

}

}

/\* POSITION SLIDES

================================================== \*/

var positionSlides = function() {

var pos = 0;

for(var i = 0; i < slides.length; i++) {

pos = i \* (config.slider.width+config.spacing);

slides[i].leftpos(pos);

}

}

/\* OPACITY SLIDES

================================================== \*/

var opacitySlides = function(n) {

var \_ease = "linear";

for(var i = 0; i < slides.length; i++) {

if (i == current\_slide) {

slides[i].animate(config.duration, \_ease, {"opacity": 1});

} else if (i == current\_slide - 1 || i == current\_slide + 1) {

slides[i].animate(config.duration, \_ease, {"opacity": 0.1});

} else {

slides[i].opacity(n);

}

}

}

/\* GO TO SLIDE

goToSlide(n, ease, duration);

================================================== \*/

var goToSlide = function(n, ease, duration, fast, firstrun) {

/\* STOP ANY VIDEO PLAYERS ACTIVE

================================================== \*/

VMM.ExternalAPI.youtube.stopPlayers();

// Set current slide

current\_slide = n;

var \_ease = config.ease;

var \_duration = config.duration;

var is\_last = false;

var is\_first = false;

var \_pos = slides[current\_slide].leftpos();

var \_title = "";

if (current\_slide == 0) {is\_first = true};

if (current\_slide +1 >= slides.length) {is\_last = true};

if (ease != null && ease != "") {\_ease = ease};

if (duration != null && duration != "") {\_duration = duration};

/\* set proper nav titles and dates etc.

================================================== \*/

if (is\_first) {

VMM.Lib.visible(navigation.prevBtn, false);

} else {

VMM.Lib.visible(navigation.prevBtn, true);

\_title = VMM.Util.unlinkify(data[current\_slide - 1].title)

if (config.type == "timeline") {

if(typeof data[current\_slide - 1].date === "undefined") {

VMM.attachElement(navigation.prevDate, \_title);

VMM.attachElement(navigation.prevTitle, "");

} else {

VMM.attachElement(navigation.prevDate, VMM.Date.prettyDate(data[current\_slide - 1].startdate));

VMM.attachElement(navigation.prevTitle, \_title);

}

} else {

VMM.attachElement(navigation.prevTitle, \_title);

}

}

if (is\_last) {

VMM.Lib.visible(navigation.nextBtn, false);

} else {

VMM.Lib.visible(navigation.nextBtn, true);

\_title = VMM.Util.unlinkify(data[current\_slide + 1].title);

if (config.type == "timeline") {

if(typeof data[current\_slide + 1].date === "undefined") {

VMM.attachElement(navigation.nextDate, \_title);

VMM.attachElement(navigation.nextTitle, "");

} else {

VMM.attachElement(navigation.nextDate, VMM.Date.prettyDate(data[current\_slide + 1].startdate) );

VMM.attachElement(navigation.nextTitle, \_title);

}

} else {

VMM.attachElement(navigation.nextTitle, \_title);

}

}

/\* ANIMATE SLIDE

================================================== \*/

if (fast) {

VMM.Lib.css($slider\_container, "left", -(\_pos - config.slider.content.padding));

} else{

VMM.Lib.stop($slider\_container);

VMM.Lib.animate($slider\_container, \_duration, \_ease, {"left": -(\_pos - config.slider.content.padding)});

}

if (firstrun) {

VMM.fireEvent(layout, "LOADED");

}

/\* SET Vertical Scoll

================================================== \*/

if (slides[current\_slide].height() > config.slider\_height) {

VMM.Lib.css(".slider", "overflow-y", "scroll" );

} else {

VMM.Lib.css(layout, "overflow-y", "hidden" );

var scroll\_height = 0;

try {

scroll\_height = VMM.Lib.prop(layout, "scrollHeight");

VMM.Lib.animate(layout, \_duration, \_ease, {scrollTop: scroll\_height - VMM.Lib.height(layout) });

}

catch(err) {

scroll\_height = VMM.Lib.height(layout);

}

}

preloadSlides();

}

/\* BUILD NAVIGATION

================================================== \*/

var buildNavigation = function() {

var temp\_icon = "<div class='icon'>&nbsp;</div>";

navigation.nextBtn = VMM.appendAndGetElement($slider, "<div>", "nav-next");

navigation.prevBtn = VMM.appendAndGetElement($slider, "<div>", "nav-previous");

navigation.nextBtnContainer = VMM.appendAndGetElement(navigation.nextBtn, "<div>", "nav-container", temp\_icon);

navigation.prevBtnContainer = VMM.appendAndGetElement(navigation.prevBtn, "<div>", "nav-container", temp\_icon);

if (config.type == "timeline") {

navigation.nextDate = VMM.appendAndGetElement(navigation.nextBtnContainer, "<div>", "date", "");

navigation.prevDate = VMM.appendAndGetElement(navigation.prevBtnContainer, "<div>", "date", "");

}

navigation.nextTitle = VMM.appendAndGetElement(navigation.nextBtnContainer, "<div>", "title", "Title Goes Here");

navigation.prevTitle = VMM.appendAndGetElement(navigation.prevBtnContainer, "<div>", "title", "Title Goes Here");

VMM.bindEvent(".nav-next", onNextClick);

VMM.bindEvent(".nav-previous", onPrevClick);

VMM.bindEvent(window, onKeypressNav, 'keydown');

}

/\* BUILD

================================================== \*/

var build = function() {

// Clear out existing content

VMM.attachElement(layout, "");

// Get DOM Objects to local objects

$slider = VMM.getElement("div.slider");

$slider\_mask = VMM.appendAndGetElement($slider, "<div>", "slider-container-mask");

$slider\_container = VMM.appendAndGetElement($slider\_mask, "<div>", "slider-container");

$slides\_items = VMM.appendAndGetElement($slider\_container, "<div>", "slider-item-container");

// BUILD NAVIGATION

buildNavigation();

// ATTACH SLIDES

buildSlides(data);

/\* MAKE SLIDER TOUCHABLE

================================================== \*/

var \_\_duration = 3000;

if (VMM.Browser.device == "tablet" || VMM.Browser.device == "mobile") {

config.duration = 500;

\_\_duration = 1000;

//VMM.TouchSlider.createPanel($slider\_container, $slider\_container, VMM.Lib.width(slides[0]), config.spacing, true);

//VMM.TouchSlider.createPanel($slider\_container, $slider\_container, slides[0].width(), config.spacing, true);

//VMM.bindEvent($slider\_container, onTouchUpdate, "TOUCHUPDATE");

} else if (VMM.Browser.device == "mobile") {

} else {

//VMM.DragSlider.createPanel($slider\_container, $slider\_container, VMM.Lib.width(slides[0]), config.spacing, true);

}

reSize(false, true);

VMM.Lib.visible(navigation.prevBtn, false);

goToSlide(config.current\_slide, "easeOutExpo", \_\_duration, true, true);

\_active = true;

};

};

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Slider.Slide.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Slider Slide

================================================== \*/

if (typeof VMM.Slider != 'undefined') {

VMM.Slider.Slide = function(d, \_parent) {

var $media, $text, $slide, $wrap, element, c,

data = d,

slide = {},

element = "",

media = "",

loaded = false,

preloaded = false,

is\_skinny = false,

\_enqueue = true,

\_removeque = false,

\_id = "slide\_",

timer = {pushque:"", render:"", relayout:"", remove:"", skinny:false},

times = {pushque:500, render:100, relayout:100, remove:30000};

\_id = \_id + data.uniqueid;

this.enqueue = \_enqueue;

this.id = \_id;

element = VMM.appendAndGetElement(\_parent, "<div>", "slider-item");

c = {slide:"", text: "", media: "", media\_element: "", layout: "content-container layout", has: { headline: false, text: false, media: false }};

/\* PUBLIC

================================================== \*/

this.show = function(skinny) {

\_enqueue = false;

timer.skinny = skinny;

\_removeque = false;

clearTimeout(timer.remove);

if (!loaded) {

if (preloaded) {

clearTimeout(timer.relayout);

timer.relayout = setTimeout(reloadLayout, times.relayout);

} else {

render(skinny);

}

}

};

this.hide = function() {

if (loaded && !\_removeque) {

\_removeque = true;

clearTimeout(timer.remove);

timer.remove = setTimeout(removeSlide, times.remove);

}

};

this.clearTimers = function() {

//clearTimeout(timer.remove);

clearTimeout(timer.relayout);

clearTimeout(timer.pushque);

clearTimeout(timer.render);

};

this.layout = function(skinny) {

if (loaded && preloaded) {

reLayout(skinny);

}

};

this.elem = function() {

return element;

};

this.position = function() {

return VMM.Lib.position(element);

};

this.leftpos = function(p) {

if(typeof p != 'undefined') {

VMM.Lib.css(element, "left", p);

} else {

return VMM.Lib.position(element).left

}

};

this.animate = function(d, e, p) {

VMM.Lib.animate(element, d, e, p);

};

this.css = function(p, v) {

VMM.Lib.css(element, p, v );

}

this.opacity = function(p) {

VMM.Lib.css(element, "opacity", p);

}

this.width = function() {

return VMM.Lib.width(element);

};

this.height = function() {

return VMM.Lib.height(element);

};

this.content\_height = function () {

var ch = VMM.Lib.find( element, ".content")[0];

if (ch != 'undefined' && ch != null) {

return VMM.Lib.height(ch);

} else {

return 0;

}

}

/\* PRIVATE

================================================== \*/

var render = function(skinny) {

trace("RENDER " + \_id);

loaded = true;

preloaded = true;

timer.skinny = skinny;

buildSlide();

clearTimeout(timer.pushque);

clearTimeout(timer.render);

timer.pushque = setTimeout(VMM.ExternalAPI.pushQues, times.pushque);

};

var removeSlide = function() {

//VMM.attachElement(element, "");

trace("REMOVE SLIDE TIMER FINISHED");

loaded = false;

VMM.Lib.detach($text);

VMM.Lib.detach($media);

};

var reloadLayout = function() {

loaded = true;

reLayout(timer.skinny, true);

};

var reLayout = function(skinny, reload) {

if (c.has.text) {

if (skinny) {

if (!is\_skinny || reload) {

VMM.Lib.removeClass($slide, "pad-left");

VMM.Lib.detach($text);

VMM.Lib.detach($media);

VMM.Lib.append($slide, $text);

VMM.Lib.append($slide, $media);

is\_skinny = true;

}

} else {

if (is\_skinny || reload) {

VMM.Lib.addClass($slide, "pad-left");

VMM.Lib.detach($text);

VMM.Lib.detach($media);

VMM.Lib.append($slide, $media);

VMM.Lib.append($slide, $text);

is\_skinny = false;

}

}

} else if (reload) {

if (c.has.headline) {

VMM.Lib.detach($text);

VMM.Lib.append($slide, $text);

}

VMM.Lib.detach($media);

VMM.Lib.append($slide, $media);

}

}

var buildSlide = function() {

trace("BUILDSLIDE");

$wrap = VMM.appendAndGetElement(element, "<div>", "content");

$slide = VMM.appendAndGetElement($wrap, "<div>");

/\* DATE

================================================== \*/

if (data.startdate != null && data.startdate != "") {

if (type.of(data.startdate) == "date") {

if (data.type != "start") {

var st = VMM.Date.prettyDate(data.startdate);

var en = VMM.Date.prettyDate(data.enddate);

var tag = "";

/\* TAG / CATEGORY

================================================== \*/

if (data.tag != null && data.tag != "") {

tag = VMM.createElement("span", data.tag, "slide-tag");

}

if (st != en) {

c.text += VMM.createElement("h2", st + " &mdash; " + en + tag, "date");

} else {

c.text += VMM.createElement("h2", st + tag, "date");

}

}

}

}

/\* HEADLINE

================================================== \*/

if (data.headline != null && data.headline != "") {

c.has.headline = true;

if (data.type == "start") {

c.text += VMM.createElement("h2", VMM.Util.linkify\_with\_twitter(data.headline, "\_blank"), "start");

} else {

c.text += VMM.createElement("h3", VMM.Util.linkify\_with\_twitter(data.headline, "\_blank"));

}

}

/\* TEXT

================================================== \*/

if (data.text != null && data.text != "") {

c.has.text = true;

c.text += VMM.createElement("p", VMM.Util.linkify\_with\_twitter(data.text, "\_blank"));

}

if (c.has.text || c.has.headline) {

c.text = VMM.createElement("div", c.text, "container");

//$text = VMM.appendAndGetElement($slide, "<div>", "text", c.text);

$text = VMM.appendAndGetElement($slide, "<div>", "text", VMM.TextElement.create(c.text));

}

/\* SLUG

================================================== \*/

if (data.needs\_slug) {

}

/\* MEDIA

================================================== \*/

if (data.asset != null && data.asset != "") {

if (data.asset.media != null && data.asset.media != "") {

c.has.media = true;

$media = VMM.appendAndGetElement($slide, "<div>", "media", VMM.MediaElement.create(data.asset, data.uniqueid));

}

}

/\* COMBINE

================================================== \*/

if (c.has.text) { c.layout += "-text" };

if (c.has.media){ c.layout += "-media" };

if (c.has.text) {

if (timer.skinny) {

VMM.Lib.addClass($slide, c.layout);

is\_skinny = true;

} else {

VMM.Lib.addClass($slide, c.layout);

VMM.Lib.addClass($slide, "pad-left");

VMM.Lib.detach($text);

VMM.Lib.append($slide, $text);

}

} else {

VMM.Lib.addClass($slide, c.layout);

}

};

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Language.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* DEFAULT LANGUAGE

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Language == 'undefined') {

VMM.Language = {

lang: "en",

api: {

wikipedia: "en"

},

date: {

month: ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"],

month\_abbr: ["Jan.", "Feb.", "March", "April", "May", "June", "July", "Aug.", "Sept.", "Oct.", "Nov.", "Dec."],

day: ["Sunday","Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"],

day\_abbr: ["Sun.","Mon.", "Tues.", "Wed.", "Thurs.", "Fri.", "Sat."]

},

dateformats: {

year: "yyyy",

month\_short: "mmm",

month: "mmmm yyyy",

full\_short: "mmm d",

full: "mmmm d',' yyyy",

time\_no\_seconds\_short: "h:MM TT",

time\_no\_seconds\_small\_date: "h:MM TT'<br/><small>'mmmm d',' yyyy'</small>'",

full\_long: "mmm d',' yyyy 'at' hh:MM TT",

full\_long\_small\_date: "hh:MM TT'<br/><small>mmm d',' yyyy'</small>'"

},

messages: {

loading\_timeline: "Loading Timeline... ",

return\_to\_title: "Return to Title",

expand\_timeline: "Expand Timeline",

contract\_timeline: "Contract Timeline",

wikipedia: "From Wikipedia, the free encyclopedia",

loading\_content: "Loading Content",

loading: "Loading"

}

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin AES.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

/\* AES implementation in JavaScript (c) Chris Veness 2005-2011 \*/

/\* - see http://csrc.nist.gov/publications/PubsFIPS.html#197 \*/

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

var Aes = {}; // Aes namespace

/\*\*

\* AES Cipher function: encrypt 'input' state with Rijndael algorithm

\* applies Nr rounds (10/12/14) using key schedule w for 'add round key' stage

\*

\* @param {Number[]} input 16-byte (128-bit) input state array

\* @param {Number[][]} w Key schedule as 2D byte-array (Nr+1 x Nb bytes)

\* @returns {Number[]} Encrypted output state array

\*/

Aes.cipher = function(input, w) { // main Cipher function [§5.1]

var Nb = 4; // block size (in words): no of columns in state (fixed at 4 for AES)

var Nr = w.length/Nb - 1; // no of rounds: 10/12/14 for 128/192/256-bit keys

var state = [[],[],[],[]]; // initialise 4xNb byte-array 'state' with input [§3.4]

for (var i=0; i<4\*Nb; i++) state[i%4][Math.floor(i/4)] = input[i];

state = Aes.addRoundKey(state, w, 0, Nb);

for (var round=1; round<Nr; round++) {

state = Aes.subBytes(state, Nb);

state = Aes.shiftRows(state, Nb);

state = Aes.mixColumns(state, Nb);

state = Aes.addRoundKey(state, w, round, Nb);

}

state = Aes.subBytes(state, Nb);

state = Aes.shiftRows(state, Nb);

state = Aes.addRoundKey(state, w, Nr, Nb);

var output = new Array(4\*Nb); // convert state to 1-d array before returning [§3.4]

for (var i=0; i<4\*Nb; i++) output[i] = state[i%4][Math.floor(i/4)];

return output;

}

/\*\*

\* Perform Key Expansion to generate a Key Schedule

\*

\* @param {Number[]} key Key as 16/24/32-byte array

\* @returns {Number[][]} Expanded key schedule as 2D byte-array (Nr+1 x Nb bytes)

\*/

Aes.keyExpansion = function(key) { // generate Key Schedule (byte-array Nr+1 x Nb) from Key [§5.2]

var Nb = 4; // block size (in words): no of columns in state (fixed at 4 for AES)

var Nk = key.length/4 // key length (in words): 4/6/8 for 128/192/256-bit keys

var Nr = Nk + 6; // no of rounds: 10/12/14 for 128/192/256-bit keys

var w = new Array(Nb\*(Nr+1));

var temp = new Array(4);

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

var r = [key[4\*i], key[4\*i+1], key[4\*i+2], key[4\*i+3]];

w[i] = r;

}

for (var i=Nk; i<(Nb\*(Nr+1)); i++) {

w[i] = new Array(4);

for (var t=0; t<4; t++) temp[t] = w[i-1][t];

if (i % Nk == 0) {

temp = Aes.subWord(Aes.rotWord(temp));

for (var t=0; t<4; t++) temp[t] ^= Aes.rCon[i/Nk][t];

} else if (Nk > 6 && i%Nk == 4) {

temp = Aes.subWord(temp);

}

for (var t=0; t<4; t++) w[i][t] = w[i-Nk][t] ^ temp[t];

}

return w;

}

/\*

\* ---- remaining routines are private, not called externally ----

\*/

Aes.subBytes = function(s, Nb) { // apply SBox to state S [§5.1.1]

for (var r=0; r<4; r++) {

for (var c=0; c<Nb; c++) s[r][c] = Aes.sBox[s[r][c]];

}

return s;

}

Aes.shiftRows = function(s, Nb) { // shift row r of state S left by r bytes [§5.1.2]

var t = new Array(4);

for (var r=1; r<4; r++) {

for (var c=0; c<4; c++) t[c] = s[r][(c+r)%Nb]; // shift into temp copy

for (var c=0; c<4; c++) s[r][c] = t[c]; // and copy back

} // note that this will work for Nb=4,5,6, but not 7,8 (always 4 for AES):

return s; // see asmaes.sourceforge.net/rijndael/rijndaelImplementation.pdf

}

Aes.mixColumns = function(s, Nb) { // combine bytes of each col of state S [§5.1.3]

for (var c=0; c<4; c++) {

var a = new Array(4); // 'a' is a copy of the current column from 's'

var b = new Array(4); // 'b' is a•{02} in GF(2^8)

for (var i=0; i<4; i++) {

a[i] = s[i][c];

b[i] = s[i][c]&0x80 ? s[i][c]<<1 ^ 0x011b : s[i][c]<<1;

}

// a[n] ^ b[n] is a•{03} in GF(2^8)

s[0][c] = b[0] ^ a[1] ^ b[1] ^ a[2] ^ a[3]; // 2\*a0 + 3\*a1 + a2 + a3

s[1][c] = a[0] ^ b[1] ^ a[2] ^ b[2] ^ a[3]; // a0 \* 2\*a1 + 3\*a2 + a3

s[2][c] = a[0] ^ a[1] ^ b[2] ^ a[3] ^ b[3]; // a0 + a1 + 2\*a2 + 3\*a3

s[3][c] = a[0] ^ b[0] ^ a[1] ^ a[2] ^ b[3]; // 3\*a0 + a1 + a2 + 2\*a3

}

return s;

}

Aes.addRoundKey = function(state, w, rnd, Nb) { // xor Round Key into state S [§5.1.4]

for (var r=0; r<4; r++) {

for (var c=0; c<Nb; c++) state[r][c] ^= w[rnd\*4+c][r];

}

return state;

}

Aes.subWord = function(w) { // apply SBox to 4-byte word w

for (var i=0; i<4; i++) w[i] = Aes.sBox[w[i]];

return w;

}

Aes.rotWord = function(w) { // rotate 4-byte word w left by one byte

var tmp = w[0];

for (var i=0; i<3; i++) w[i] = w[i+1];

w[3] = tmp;

return w;

}

// sBox is pre-computed multiplicative inverse in GF(2^8) used in subBytes and keyExpansion [§5.1.1]

Aes.sBox = [0x63,0x7c,0x77,0x7b,0xf2,0x6b,0x6f,0xc5,0x30,0x01,0x67,0x2b,0xfe,0xd7,0xab,0x76,

0xca,0x82,0xc9,0x7d,0xfa,0x59,0x47,0xf0,0xad,0xd4,0xa2,0xaf,0x9c,0xa4,0x72,0xc0,

0xb7,0xfd,0x93,0x26,0x36,0x3f,0xf7,0xcc,0x34,0xa5,0xe5,0xf1,0x71,0xd8,0x31,0x15,

0x04,0xc7,0x23,0xc3,0x18,0x96,0x05,0x9a,0x07,0x12,0x80,0xe2,0xeb,0x27,0xb2,0x75,

0x09,0x83,0x2c,0x1a,0x1b,0x6e,0x5a,0xa0,0x52,0x3b,0xd6,0xb3,0x29,0xe3,0x2f,0x84,

0x53,0xd1,0x00,0xed,0x20,0xfc,0xb1,0x5b,0x6a,0xcb,0xbe,0x39,0x4a,0x4c,0x58,0xcf,

0xd0,0xef,0xaa,0xfb,0x43,0x4d,0x33,0x85,0x45,0xf9,0x02,0x7f,0x50,0x3c,0x9f,0xa8,

0x51,0xa3,0x40,0x8f,0x92,0x9d,0x38,0xf5,0xbc,0xb6,0xda,0x21,0x10,0xff,0xf3,0xd2,

0xcd,0x0c,0x13,0xec,0x5f,0x97,0x44,0x17,0xc4,0xa7,0x7e,0x3d,0x64,0x5d,0x19,0x73,

0x60,0x81,0x4f,0xdc,0x22,0x2a,0x90,0x88,0x46,0xee,0xb8,0x14,0xde,0x5e,0x0b,0xdb,

0xe0,0x32,0x3a,0x0a,0x49,0x06,0x24,0x5c,0xc2,0xd3,0xac,0x62,0x91,0x95,0xe4,0x79,

0xe7,0xc8,0x37,0x6d,0x8d,0xd5,0x4e,0xa9,0x6c,0x56,0xf4,0xea,0x65,0x7a,0xae,0x08,

0xba,0x78,0x25,0x2e,0x1c,0xa6,0xb4,0xc6,0xe8,0xdd,0x74,0x1f,0x4b,0xbd,0x8b,0x8a,

0x70,0x3e,0xb5,0x66,0x48,0x03,0xf6,0x0e,0x61,0x35,0x57,0xb9,0x86,0xc1,0x1d,0x9e,

0xe1,0xf8,0x98,0x11,0x69,0xd9,0x8e,0x94,0x9b,0x1e,0x87,0xe9,0xce,0x55,0x28,0xdf,

0x8c,0xa1,0x89,0x0d,0xbf,0xe6,0x42,0x68,0x41,0x99,0x2d,0x0f,0xb0,0x54,0xbb,0x16];

// rCon is Round Constant used for the Key Expansion [1st col is 2^(r-1) in GF(2^8)] [§5.2]

Aes.rCon = [ [0x00, 0x00, 0x00, 0x00],

[0x01, 0x00, 0x00, 0x00],

[0x02, 0x00, 0x00, 0x00],

[0x04, 0x00, 0x00, 0x00],

[0x08, 0x00, 0x00, 0x00],

[0x10, 0x00, 0x00, 0x00],

[0x20, 0x00, 0x00, 0x00],

[0x40, 0x00, 0x00, 0x00],

[0x80, 0x00, 0x00, 0x00],

[0x1b, 0x00, 0x00, 0x00],

[0x36, 0x00, 0x00, 0x00] ];

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

/\* AES Counter-mode implementation in JavaScript (c) Chris Veness 2005-2011 \*/

/\* - see http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf \*/

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

Aes.Ctr = {}; // Aes.Ctr namespace: a subclass or extension of Aes

/\*\*

\* Encrypt a text using AES encryption in Counter mode of operation

\*

\* Unicode multi-byte character safe

\*

\* @param {String} plaintext Source text to be encrypted

\* @param {String} password The password to use to generate a key

\* @param {Number} nBits Number of bits to be used in the key (128, 192, or 256)

\* @returns {string} Encrypted text

\*/

Aes.Ctr.encrypt = function(plaintext, password, nBits) {

var blockSize = 16; // block size fixed at 16 bytes / 128 bits (Nb=4) for AES

if (!(nBits==128 || nBits==192 || nBits==256)) return ''; // standard allows 128/192/256 bit keys

plaintext = Utf8.encode(plaintext);

password = Utf8.encode(password);

//var t = new Date(); // timer

// use AES itself to encrypt password to get cipher key (using plain password as source for key

// expansion) - gives us well encrypted key (though hashed key might be preferred for prod'n use)

var nBytes = nBits/8; // no bytes in key (16/24/32)

var pwBytes = new Array(nBytes);

for (var i=0; i<nBytes; i++) { // use 1st 16/24/32 chars of password for key

pwBytes[i] = isNaN(password.charCodeAt(i)) ? 0 : password.charCodeAt(i);

}

var key = Aes.cipher(pwBytes, Aes.keyExpansion(pwBytes)); // gives us 16-byte key

key = key.concat(key.slice(0, nBytes-16)); // expand key to 16/24/32 bytes long

// initialise 1st 8 bytes of counter block with nonce (NIST SP800-38A §B.2): [0-1] = millisec,

// [2-3] = random, [4-7] = seconds, together giving full sub-millisec uniqueness up to Feb 2106

var counterBlock = new Array(blockSize);

var nonce = (new Date()).getTime(); // timestamp: milliseconds since 1-Jan-1970

var nonceMs = nonce%1000;

var nonceSec = Math.floor(nonce/1000);

var nonceRnd = Math.floor(Math.random()\*0xffff);

for (var i=0; i<2; i++) counterBlock[i] = (nonceMs >>> i\*8) & 0xff;

for (var i=0; i<2; i++) counterBlock[i+2] = (nonceRnd >>> i\*8) & 0xff;

for (var i=0; i<4; i++) counterBlock[i+4] = (nonceSec >>> i\*8) & 0xff;

// and convert it to a string to go on the front of the ciphertext

var ctrTxt = '';

for (var i=0; i<8; i++) ctrTxt += String.fromCharCode(counterBlock[i]);

// generate key schedule - an expansion of the key into distinct Key Rounds for each round

var keySchedule = Aes.keyExpansion(key);

var blockCount = Math.ceil(plaintext.length/blockSize);

var ciphertxt = new Array(blockCount); // ciphertext as array of strings

for (var b=0; b<blockCount; b++) {

// set counter (block #) in last 8 bytes of counter block (leaving nonce in 1st 8 bytes)

// done in two stages for 32-bit ops: using two words allows us to go past 2^32 blocks (68GB)

for (var c=0; c<4; c++) counterBlock[15-c] = (b >>> c\*8) & 0xff;

for (var c=0; c<4; c++) counterBlock[15-c-4] = (b/0x100000000 >>> c\*8)

var cipherCntr = Aes.cipher(counterBlock, keySchedule); // -- encrypt counter block --

// block size is reduced on final block

var blockLength = b<blockCount-1 ? blockSize : (plaintext.length-1)%blockSize+1;

var cipherChar = new Array(blockLength);

for (var i=0; i<blockLength; i++) { // -- xor plaintext with ciphered counter char-by-char --

cipherChar[i] = cipherCntr[i] ^ plaintext.charCodeAt(b\*blockSize+i);

cipherChar[i] = String.fromCharCode(cipherChar[i]);

}

ciphertxt[b] = cipherChar.join('');

}

// Array.join is more efficient than repeated string concatenation in IE

var ciphertext = ctrTxt + ciphertxt.join('');

ciphertext = Base64.encode(ciphertext); // encode in base64

//alert((new Date()) - t);

return ciphertext;

}

/\*\*

\* Decrypt a text encrypted by AES in counter mode of operation

\*

\* @param {String} ciphertext Source text to be encrypted

\* @param {String} password The password to use to generate a key

\* @param {Number} nBits Number of bits to be used in the key (128, 192, or 256)

\* @returns {String} Decrypted text

\*/

Aes.Ctr.decrypt = function(ciphertext, password, nBits) {

var blockSize = 16; // block size fixed at 16 bytes / 128 bits (Nb=4) for AES

if (!(nBits==128 || nBits==192 || nBits==256)) return ''; // standard allows 128/192/256 bit keys

ciphertext = Base64.decode(ciphertext);

password = Utf8.encode(password);

//var t = new Date(); // timer

// use AES to encrypt password (mirroring encrypt routine)

var nBytes = nBits/8; // no bytes in key

var pwBytes = new Array(nBytes);

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

pwBytes[i] = isNaN(password.charCodeAt(i)) ? 0 : password.charCodeAt(i);

}

var key = Aes.cipher(pwBytes, Aes.keyExpansion(pwBytes));

key = key.concat(key.slice(0, nBytes-16)); // expand key to 16/24/32 bytes long

// recover nonce from 1st 8 bytes of ciphertext

var counterBlock = new Array(8);

ctrTxt = ciphertext.slice(0, 8);

for (var i=0; i<8; i++) counterBlock[i] = ctrTxt.charCodeAt(i);

// generate key schedule

var keySchedule = Aes.keyExpansion(key);

// separate ciphertext into blocks (skipping past initial 8 bytes)

var nBlocks = Math.ceil((ciphertext.length-8) / blockSize);

var ct = new Array(nBlocks);

for (var b=0; b<nBlocks; b++) ct[b] = ciphertext.slice(8+b\*blockSize, 8+b\*blockSize+blockSize);

ciphertext = ct; // ciphertext is now array of block-length strings

// plaintext will get generated block-by-block into array of block-length strings

var plaintxt = new Array(ciphertext.length);

for (var b=0; b<nBlocks; b++) {

// set counter (block #) in last 8 bytes of counter block (leaving nonce in 1st 8 bytes)

for (var c=0; c<4; c++) counterBlock[15-c] = ((b) >>> c\*8) & 0xff;

for (var c=0; c<4; c++) counterBlock[15-c-4] = (((b+1)/0x100000000-1) >>> c\*8) & 0xff;

var cipherCntr = Aes.cipher(counterBlock, keySchedule); // encrypt counter block

var plaintxtByte = new Array(ciphertext[b].length);

for (var i=0; i<ciphertext[b].length; i++) {

// -- xor plaintxt with ciphered counter byte-by-byte --

plaintxtByte[i] = cipherCntr[i] ^ ciphertext[b].charCodeAt(i);

plaintxtByte[i] = String.fromCharCode(plaintxtByte[i]);

}

plaintxt[b] = plaintxtByte.join('');

}

// join array of blocks into single plaintext string

var plaintext = plaintxt.join('');

plaintext = Utf8.decode(plaintext); // decode from UTF8 back to Unicode multi-byte chars

//alert((new Date()) - t);

return plaintext;

}

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

/\* Base64 class: Base 64 encoding / decoding (c) Chris Veness 2002-2011 \*/

/\* note: depends on Utf8 class \*/

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

var Base64 = {}; // Base64 namespace

Base64.code = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";

/\*\*

\* Encode string into Base64, as defined by RFC 4648 [http://tools.ietf.org/html/rfc4648]

\* (instance method extending String object). As per RFC 4648, no newlines are added.

\*

\* @param {String} str The string to be encoded as base-64

\* @param {Boolean} [utf8encode=false] Flag to indicate whether str is Unicode string to be encoded

\* to UTF8 before conversion to base64; otherwise string is assumed to be 8-bit characters

\* @returns {String} Base64-encoded string

\*/

Base64.encode = function(str, utf8encode) { // http://tools.ietf.org/html/rfc4648

utf8encode = (typeof utf8encode == 'undefined') ? false : utf8encode;

var o1, o2, o3, bits, h1, h2, h3, h4, e=[], pad = '', c, plain, coded;

var b64 = Base64.code;

plain = utf8encode ? str.encodeUTF8() : str;

c = plain.length % 3; // pad string to length of multiple of 3

if (c > 0) { while (c++ < 3) { pad += '='; plain += '\0'; } }

// note: doing padding here saves us doing special-case packing for trailing 1 or 2 chars

for (c=0; c<plain.length; c+=3) { // pack three octets into four hexets

o1 = plain.charCodeAt(c);

o2 = plain.charCodeAt(c+1);

o3 = plain.charCodeAt(c+2);

bits = o1<<16 | o2<<8 | o3;

h1 = bits>>18 & 0x3f;

h2 = bits>>12 & 0x3f;

h3 = bits>>6 & 0x3f;

h4 = bits & 0x3f;

// use hextets to index into code string

e[c/3] = b64.charAt(h1) + b64.charAt(h2) + b64.charAt(h3) + b64.charAt(h4);

}

coded = e.join(''); // join() is far faster than repeated string concatenation in IE

// replace 'A's from padded nulls with '='s

coded = coded.slice(0, coded.length-pad.length) + pad;

return coded;

}

/\*\*

\* Decode string from Base64, as defined by RFC 4648 [http://tools.ietf.org/html/rfc4648]

\* (instance method extending String object). As per RFC 4648, newlines are not catered for.

\*

\* @param {String} str The string to be decoded from base-64

\* @param {Boolean} [utf8decode=false] Flag to indicate whether str is Unicode string to be decoded

\* from UTF8 after conversion from base64

\* @returns {String} decoded string

\*/

Base64.decode = function(str, utf8decode) {

utf8decode = (typeof utf8decode == 'undefined') ? false : utf8decode;

var o1, o2, o3, h1, h2, h3, h4, bits, d=[], plain, coded;

var b64 = Base64.code;

coded = utf8decode ? str.decodeUTF8() : str;

for (var c=0; c<coded.length; c+=4) { // unpack four hexets into three octets

h1 = b64.indexOf(coded.charAt(c));

h2 = b64.indexOf(coded.charAt(c+1));

h3 = b64.indexOf(coded.charAt(c+2));

h4 = b64.indexOf(coded.charAt(c+3));

bits = h1<<18 | h2<<12 | h3<<6 | h4;

o1 = bits>>>16 & 0xff;

o2 = bits>>>8 & 0xff;

o3 = bits & 0xff;

d[c/4] = String.fromCharCode(o1, o2, o3);

// check for padding

if (h4 == 0x40) d[c/4] = String.fromCharCode(o1, o2);

if (h3 == 0x40) d[c/4] = String.fromCharCode(o1);

}

plain = d.join(''); // join() is far faster than repeated string concatenation in IE

return utf8decode ? plain.decodeUTF8() : plain;

}

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

/\* Utf8 class: encode / decode between multi-byte Unicode characters and UTF-8 multiple \*/

/\* single-byte character encoding (c) Chris Veness 2002-2011 \*/

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

var Utf8 = {}; // Utf8 namespace

/\*\*

\* Encode multi-byte Unicode string into utf-8 multiple single-byte characters

\* (BMP / basic multilingual plane only)

\*

\* Chars in range U+0080 - U+07FF are encoded in 2 chars, U+0800 - U+FFFF in 3 chars

\*

\* @param {String} strUni Unicode string to be encoded as UTF-8

\* @returns {String} encoded string

\*/

Utf8.encode = function(strUni) {

// use regular expressions & String.replace callback function for better efficiency

// than procedural approaches

var strUtf = strUni.replace(

/[\u0080-\u07ff]/g, // U+0080 - U+07FF => 2 bytes 110yyyyy, 10zzzzzz

function(c) {

var cc = c.charCodeAt(0);

return String.fromCharCode(0xc0 | cc>>6, 0x80 | cc&0x3f); }

);

strUtf = strUtf.replace(

/[\u0800-\uffff]/g, // U+0800 - U+FFFF => 3 bytes 1110xxxx, 10yyyyyy, 10zzzzzz

function(c) {

var cc = c.charCodeAt(0);

return String.fromCharCode(0xe0 | cc>>12, 0x80 | cc>>6&0x3F, 0x80 | cc&0x3f); }

);

return strUtf;

}

/\*\*

\* Decode utf-8 encoded string back into multi-byte Unicode characters

\*

\* @param {String} strUtf UTF-8 string to be decoded back to Unicode

\* @returns {String} decoded string

\*/

Utf8.decode = function(strUtf) {

// note: decode 3-byte chars first as decoded 2-byte strings could appear to be 3-byte char!

var strUni = strUtf.replace(

/[\u00e0-\u00ef][\u0080-\u00bf][\u0080-\u00bf]/g, // 3-byte chars

function(c) { // (note parentheses for precence)

var cc = ((c.charCodeAt(0)&0x0f)<<12) | ((c.charCodeAt(1)&0x3f)<<6) | ( c.charCodeAt(2)&0x3f);

return String.fromCharCode(cc); }

);

strUni = strUni.replace(

/[\u00c0-\u00df][\u0080-\u00bf]/g, // 2-byte chars

function(c) { // (note parentheses for precence)

var cc = (c.charCodeAt(0)&0x1f)<<6 | c.charCodeAt(1)&0x3f;

return String.fromCharCode(cc); }

);

return strUni;

}

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin bootstrap-tooltip.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* ===========================================================

\* bootstrap-tooltip.js v2.0.1

\* http://twitter.github.com/bootstrap/javascript.html#tooltips

\* Inspired by the original jQuery.tipsy by Jason Frame

\* ===========================================================

\* Copyright 2012 Twitter, Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\* ========================================================== \*/

!function( $ ) {

"use strict"

/\* TOOLTIP PUBLIC CLASS DEFINITION

\* =============================== \*/

var Tooltip = function ( element, options ) {

this.init('tooltip', element, options)

}

Tooltip.prototype = {

constructor: Tooltip

, init: function ( type, element, options ) {

var eventIn

, eventOut

this.type = type

this.$element = $(element)

this.options = this.getOptions(options)

this.enabled = true

if (this.options.trigger != 'manual') {

eventIn = this.options.trigger == 'hover' ? 'mouseenter' : 'focus'

eventOut = this.options.trigger == 'hover' ? 'mouseleave' : 'blur'

this.$element.on(eventIn, this.options.selector, $.proxy(this.enter, this))

this.$element.on(eventOut, this.options.selector, $.proxy(this.leave, this))

}

this.options.selector ?

(this.\_options = $.extend({}, this.options, { trigger: 'manual', selector: '' })) :

this.fixTitle()

}

, getOptions: function ( options ) {

options = $.extend({}, $.fn[this.type].defaults, options, this.$element.data())

if (options.delay && typeof options.delay == 'number') {

options.delay = {

show: options.delay

, hide: options.delay

}

}

return options

}

, enter: function ( e ) {

var self = $(e.currentTarget)[this.type](this.\_options).data(this.type)

if (!self.options.delay || !self.options.delay.show) {

self.show()

} else {

self.hoverState = 'in'

setTimeout(function() {

if (self.hoverState == 'in') {

self.show()

}

}, self.options.delay.show)

}

}

, leave: function ( e ) {

var self = $(e.currentTarget)[this.type](this.\_options).data(this.type)

if (!self.options.delay || !self.options.delay.hide) {

self.hide()

} else {

self.hoverState = 'out'

setTimeout(function() {

if (self.hoverState == 'out') {

self.hide()

}

}, self.options.delay.hide)

}

}

, show: function () {

var $tip

, inside

, pos

, actualWidth

, actualHeight

, placement

, tp

if (this.hasContent() && this.enabled) {

$tip = this.tip()

this.setContent()

if (this.options.animation) {

$tip.addClass('fade')

}

placement = typeof this.options.placement == 'function' ?

this.options.placement.call(this, $tip[0], this.$element[0]) :

this.options.placement

inside = /in/.test(placement)

$tip

.remove()

.css({ top: 0, left: 0, display: 'block' })

.appendTo(inside ? this.$element : document.body)

pos = this.getPosition(inside)

actualWidth = $tip[0].offsetWidth

actualHeight = $tip[0].offsetHeight

switch (inside ? placement.split(' ')[1] : placement) {

case 'bottom':

tp = {top: pos.top + pos.height, left: pos.left + pos.width / 2 - actualWidth / 2}

break

case 'top':

tp = {top: pos.top - actualHeight, left: pos.left + pos.width / 2 - actualWidth / 2}

break

case 'left':

tp = {top: pos.top + pos.height / 2 - actualHeight / 2, left: pos.left - actualWidth}

break

case 'right':

tp = {top: pos.top + pos.height / 2 - actualHeight / 2, left: pos.left + pos.width}

break

}

$tip

.css(tp)

.addClass(placement)

.addClass('in')

}

}

, setContent: function () {

var $tip = this.tip()

$tip.find('.tooltip-inner').html(this.getTitle())

$tip.removeClass('fade in top bottom left right')

}

, hide: function () {

var that = this

, $tip = this.tip()

$tip.removeClass('in')

function removeWithAnimation() {

var timeout = setTimeout(function () {

$tip.off($.support.transition.end).remove()

}, 500)

$tip.one($.support.transition.end, function () {

clearTimeout(timeout)

$tip.remove()

})

}

$.support.transition && this.$tip.hasClass('fade') ?

removeWithAnimation() :

$tip.remove()

}

, fixTitle: function () {

var $e = this.$element

if ($e.attr('title') || typeof($e.attr('data-original-title')) != 'string') {

$e.attr('data-original-title', $e.attr('title') || '').removeAttr('title')

}

}

, hasContent: function () {

return this.getTitle()

}

, getPosition: function (inside) {

return $.extend({}, (inside ? {top: 0, left: 0} : this.$element.offset()), {

width: this.$element[0].offsetWidth

, height: this.$element[0].offsetHeight

})

}

, getTitle: function () {

var title

, $e = this.$element

, o = this.options

title = $e.attr('data-original-title')

|| (typeof o.title == 'function' ? o.title.call($e[0]) : o.title)

title = title.toString().replace(/(^\s\*|\s\*$)/, "")

return title

}

, tip: function () {

return this.$tip = this.$tip || $(this.options.template)

}

, validate: function () {

if (!this.$element[0].parentNode) {

this.hide()

this.$element = null

this.options = null

}

}

, enable: function () {

this.enabled = true

}

, disable: function () {

this.enabled = false

}

, toggleEnabled: function () {

this.enabled = !this.enabled

}

, toggle: function () {

this[this.tip().hasClass('in') ? 'hide' : 'show']()

}

}

/\* TOOLTIP PLUGIN DEFINITION

\* ========================= \*/

$.fn.tooltip = function ( option ) {

return this.each(function () {

var $this = $(this)

, data = $this.data('tooltip')

, options = typeof option == 'object' && option

if (!data) $this.data('tooltip', (data = new Tooltip(this, options)))

if (typeof option == 'string') data[option]()

})

}

$.fn.tooltip.Constructor = Tooltip

$.fn.tooltip.defaults = {

animation: true

, delay: 0

, selector: false

, placement: 'top'

, trigger: 'hover'

, title: ''

, template: '<div class="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-inner"></div></div>'

}

}( window.jQuery );

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Timeline.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* Timeline

\* Designed and built by Zach Wise at VéritéCo

\* This program is free software: you can redistribute it and/or modify

\* it under the terms of the GNU General Public License as published by

\* the Free Software Foundation, either version 3 of the License, or

\* (at your option) any later version.

\* This program is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

\* GNU General Public License for more details.

\* http://www.gnu.org/licenses/

\*/

/\* \* CodeKit Import

\* http://incident57.com/codekit/

================================================== \*/

// @codekit-prepend "VMM.Timeline.License.js";

// @codekit-prepend "Core/VMM.js";

// @codekit-prepend "Core/VMM.Library.js";

// @codekit-prepend "Core/VMM.Browser.js";

// @codekit-prepend "Core/VMM.FileExtention.js";

// @codekit-prepend "Core/VMM.Date.js";

// @codekit-prepend "Core/VMM.Util.js";

// @codekit-prepend "Core/VMM.LoadLib.js";

// @codekit-prepend "Media/VMM.ExternalAPI.js";

// @codekit-prepend "Media/VMM.MediaElement.js";

// @codekit-prepend "Media/VMM.MediaType.js";

// @codekit-prepend "Media/VMM.Media.js";

// @codekit-prepend "Media/VMM.TextElement.js";

// @codekit-prepend "Slider/VMM.DragSlider.js";

// @codekit-prepend "Slider/VMM.Slider.js";

// @codekit-prepend "Slider/VMM.Slider.Slide.js";

// @codekit-prepend "VMM.Language.js";

// @codekit-append "VMM.Timeline.TimeNav.js";

// @codekit-append "VMM.Timeline.DataObj.js";

// @codekit-prepend "lib/AES.js";

// @codekit-prepend "lib/bootstrap-tooltip.js";

/\* Timeline

================================================== \*/

if(typeof VMM != 'undefined' && typeof VMM.Timeline == 'undefined') {

VMM.Timeline = function(w, h, conf, \_timeline\_id) {

var $timeline, $feedback, slider, timenav, version, timeline\_id;

var events = {}, data = {}, \_dates = [], config = {};

var has\_width = false, has\_height = false, ie7 = false, is\_moving = false;

if (type.of(\_timeline\_id) == "string") {

timeline\_id = \_timeline\_id;

} else {

timeline\_id = "#timelinejs";

}

version = "1.68";

trace("TIMELINE VERSION " + version);

/\* CONFIG

================================================== \*/

config = {

embed: false,

events: {

data\_ready: "DATAREADY",

messege: "MESSEGE",

headline: "HEADLINE",

slide\_change: "SLIDE\_CHANGE",

resize: "resize"

},

id: timeline\_id,

type: "timeline",

touch: false,

maptype: "toner",

preload: 4,

current\_slide: 0,

hash\_bookmark: false,

start\_at\_end: false,

start\_at\_slide: 0,

start\_zoom\_adjust: 0,

start\_page: false,

api\_keys: {

google: "",

flickr: "",

twitter: ""

},

interval: 10,

something: 0,

width: 960,

height: 540,

spacing: 15,

loaded: {

slider: false,

timenav: false,

percentloaded: 0

},

nav: {

start\_page: false,

interval\_width: 200,

density: 4,

minor\_width: 0,

minor\_left: 0,

constraint: {

left: 0,

right: 0,

right\_min: 0,

right\_max: 0

},

zoom: {

adjust: 0

},

multiplier: {

current: 6,

min: .1,

max: 50

},

rows: [1, 1, 1],

width: 960,

height: 200,

marker: {

width: 150,

height: 50

}

},

feature: {

width: 960,

height: 540

},

slider: {

width: 720,

height: 400,

content: {

width: 720,

height: 400,

padding: 130

},

nav: {

width: 100,

height: 200

}

},

ease: "easeInOutExpo",

duration: 1000,

language: VMM.Language

};

if ( w != null && w != "") {

config.width = w;

has\_width = true;

}

if ( h != null && h != "") {

config.height = h;

has\_height = true;

}

if(window.location.hash) {

var hash = window.location.hash.substring(1);

if (!isNaN(hash)) {

config.current\_slide = parseInt(hash);

}

}

window.onhashchange = function () {

var hash = window.location.hash.substring(1);

if (config.hash\_bookmark) {

if (is\_moving) {

goToEvent(parseInt(hash));

} else {

is\_moving = false;

}

} else {

goToEvent(parseInt(hash));

}

}

/\* CREATE CONFIG

================================================== \*/

var createConfig = function(conf) {

// APPLY SUPPLIED CONFIG TO TIMELINE CONFIG

if (typeof embed\_config == 'object') {

timeline\_config = embed\_config;

}

if (typeof timeline\_config == 'object') {

trace("HAS TIMELINE CONFIG");

config = VMM.Util.mergeConfig(config, timeline\_config);

} else if (typeof conf == 'object') {

config = VMM.Util.mergeConfig(config, conf);

}

if (VMM.Browser.device == "mobile" || VMM.Browser.device == "tablet") {

config.touch = true;

}

config.nav.width = config.width;

config.nav.height = 200;

config.feature.width = config.width;

config.feature.height = config.height - config.nav.height;

config.nav.zoom.adjust = parseInt(config.start\_zoom\_adjust, 10);

VMM.Timeline.Config = config;

VMM.master\_config.Timeline = VMM.Timeline.Config;

this.events = config.events;

}

/\* CREATE TIMELINE STRUCTURE

================================================== \*/

var createStructure = function(w, h) {

$timeline = VMM.getElement(timeline\_id);

VMM.Lib.addClass(timeline\_id, "vmm-timeline");

if (config.touch) {

VMM.Lib.addClass(timeline\_id, "vmm-touch");

} else {

VMM.Lib.addClass(timeline\_id, "vmm-notouch");

}

$feedback = VMM.appendAndGetElement($timeline, "<div>", "feedback", "");

slider = new VMM.Slider(timeline\_id + " div.slider", config);

timenav = new VMM.Timeline.TimeNav(timeline\_id + " div.navigation");

if (!has\_width) {

config.width = VMM.Lib.width($timeline);

} else {

VMM.Lib.width($timeline, config.width);

}

if (!has\_height) {

config.height = VMM.Lib.height($timeline);

} else {

VMM.Lib.height($timeline, config.height);

}

}

/\* ON EVENT

================================================== \*/

function onDataReady(e, d) {

trace("onDataReady");

trace(d);

data = d.timeline;

if (type.of(data.era) == "array") {

} else {

data.era = [];

}

buildDates();

};

function onDatesProcessed() {

build();

}

function reSize() {

updateSize();

slider.setSize(config.feature.width, config.feature.height);

timenav.setSize(config.width, config.height);

};

function onSliderLoaded(e) {

config.loaded.slider = true;

onComponentLoaded();

};

function onComponentLoaded(e) {

config.loaded.percentloaded = config.loaded.percentloaded + 25;

if (config.loaded.slider && config.loaded.timenav) {

hideMessege();

}

}

function onTimeNavLoaded(e) {

config.loaded.timenav = true;

onComponentLoaded();

}

function onSlideUpdate(e) {

is\_moving = true;

config.current\_slide = slider.getCurrentNumber();

setHash(config.current\_slide);

timenav.setMarker(config.current\_slide, config.ease,config.duration);

};

function onMarkerUpdate(e) {

is\_moving = true;

config.current\_slide = timenav.getCurrentNumber();

setHash(config.current\_slide);

slider.setSlide(config.current\_slide);

};

var goToEvent = function(n) {

if (n <= \_dates.length - 1 && n >= 0) {

config.current\_slide = n;

slider.setSlide(config.current\_slide);

timenav.setMarker(config.current\_slide, config.ease,config.duration);

}

}

function setHash(n) {

if (config.hash\_bookmark) {

window.location.hash = "#" + n.toString();

}

}

/\* PUBLIC FUNCTIONS

================================================== \*/

this.init = function(\_data, \_timeline\_id, conf) {

if (type.of(\_timeline\_id) == "string") {

if (\_timeline\_id.match("#")) {

timeline\_id = \_timeline\_id;

} else {

timeline\_id = "#" + \_timeline\_id;

}

}

createConfig(conf);

createStructure(w,h);

trace('TIMELINE INIT');

VMM.Date.setLanguage(VMM.Timeline.Config.language);

VMM.master\_config.language = VMM.Timeline.Config.language;

$feedback = VMM.appendAndGetElement($timeline, "<div>", "feedback", "");

// EVENTS

VMM.bindEvent(global, onDataReady, config.events.data\_ready);

VMM.bindEvent(global, showMessege, config.events.messege);

VMM.fireEvent(global, config.events.messege, VMM.master\_config.language.messages.loading\_timeline);

/\* GET DATA

================================================== \*/

if (VMM.Browser.browser == "Explorer" || VMM.Browser.browser == "MSIE") {

if ( parseInt(VMM.Browser.version, 10) <= 7 ) {

ie7 = true;

}

}

if (type.of(\_data) == "string" || type.of(\_data) == "object") {

VMM.Timeline.DataObj.getData(\_data);

} else {

VMM.Timeline.DataObj.getData(VMM.getElement(timeline\_id));

}

};

this.iframeLoaded = function() {

trace("iframeLoaded");

};

this.reload = function(\_d) {

trace("loadNewDates" + \_d);

VMM.fireEvent(global, config.events.messege, VMM.master\_config.language.messages.loading\_timeline);

data = {};

VMM.Timeline.DataObj.getData(\_d);

};

/\* DATA

================================================== \*/

var getData = function(url) {

VMM.getJSON(url, function(d) {

data = VMM.Timeline.DataObj.getData(d);

VMM.fireEvent(global, config.events.data\_ready);

});

};

/\* MESSEGES

================================================== \*/

var showMessege = function(e, msg) {

trace("showMessege " + msg);

//VMM.attachElement($messege, msg);

VMM.attachElement($feedback, VMM.MediaElement.loadingmessage(msg));

};

var hideMessege = function() {

VMM.Lib.animate($feedback, config.duration, config.ease\*4, {"opacity": 0}, detachMessege);

};

var detachMessege = function() {

VMM.Lib.detach($feedback);

}

/\* BUILD DISPLAY

================================================== \*/

var build = function() {

// START AT SLIDE

if (parseInt(config.start\_at\_slide) > 0 && config.current\_slide == 0) {

config.current\_slide = parseInt(config.start\_at\_slide);

}

// START AT END

if (config.start\_at\_end && config.current\_slide == 0) {

config.current\_slide = \_dates.length - 1;

}

// IE7

if (ie7) {

ie7 = true;

VMM.fireEvent(global, config.events.messege, "Internet Explorer " + VMM.Browser.version + " is not supported by TimelineJS. Please update your browser to version 8 or higher.");

} else {

// CREATE DOM STRUCTURE

VMM.attachElement($timeline, "");

VMM.appendElement($timeline, "<div class='container main'><div class='feature'><div class='slider'></div></div><div class='navigation'></div></div>");

reSize();

VMM.bindEvent("div.slider", onSliderLoaded, "LOADED");

VMM.bindEvent("div.navigation", onTimeNavLoaded, "LOADED");

VMM.bindEvent("div.slider", onSlideUpdate, "UPDATE");

VMM.bindEvent("div.navigation", onMarkerUpdate, "UPDATE");

slider.init(\_dates);

timenav.init(\_dates, data.era);

// RESIZE EVENT LISTENERS

VMM.bindEvent(global, reSize, config.events.resize);

//VMM.bindEvent(global, function(e) {e.preventDefault()}, "touchmove");

}

};

var ie7Build = function() {

trace("IE7 or lower");

for(var i = 0; i < \_dates.length; i++) {

trace(\_dates[i]);

/\*

var st = VMM.Date.prettyDate(data.startdate);

var en = VMM.Date.prettyDate(data.enddate);

var tag = "";

if (data.tag != null && data.tag != "") {

tag = VMM.createElement("span", data.tag, "slide-tag");

}

if (st != en) {

c.text += VMM.createElement("h2", st + " &mdash; " + en + tag, "date");

} else {

c.text += VMM.createElement("h2", st + tag, "date");

}

\*/

}

};

var updateSize = function() {

trace("UPDATE SIZE");

config.width = VMM.Lib.width($timeline);

config.height = VMM.Lib.height($timeline);

config.nav.width = config.width;

config.feature.width = config.width;

if (VMM.Browser.device == "mobile") {

//config.feature.height = config.height;

} else {

//config.feature.height = config.height - config.nav.height - 3;

}

config.feature.height = config.height - config.nav.height - 3;

};

// BUILD DATE OBJECTS

var buildDates = function() {

\_dates = [];

VMM.fireEvent(global, config.events.messege, "Building Dates");

updateSize();

for(var i = 0; i < data.date.length; i++) {

if (data.date[i].startDate != null && data.date[i].startDate != "") {

var \_date = {};

// START DATE

if (data.date[i].type == "tweets") {

\_date.startdate = VMM.ExternalAPI.twitter.parseTwitterDate(data.date[i].startDate);

} else {

\_date.startdate = VMM.Date.parse(data.date[i].startDate);

}

if (!isNaN(\_date.startdate)) {

// END DATE

if (data.date[i].endDate != null && data.date[i].endDate != "") {

if (data.date[i].type == "tweets") {

\_date.enddate = VMM.ExternalAPI.twitter.parseTwitterDate(data.date[i].endDate);

} else {

\_date.enddate = VMM.Date.parse(data.date[i].endDate);

}

} else {

\_date.enddate = \_date.startdate;

}

\_date.needs\_slug = false;

if (data.date[i].headline == "") {

if (data.date[i].slug != null && data.date[i].slug != "") {

\_date.needs\_slug = true;

}

}

\_date.title = data.date[i].headline;

\_date.headline = data.date[i].headline;

\_date.type = data.date[i].type;

\_date.date = VMM.Date.prettyDate(\_date.startdate);

\_date.asset = data.date[i].asset;

\_date.fulldate = \_date.startdate.getTime();

\_date.text = data.date[i].text;

\_date.content = "";

\_date.tag = data.date[i].tag;

\_date.slug = data.date[i].slug;

\_date.uniqueid = VMM.Util.unique\_ID(7);

\_dates.push(\_date);

}

}

};

/\* CUSTOM SORT

================================================== \*/

if (data.type != "storify") {

\_dates.sort(function(a, b){

return a.fulldate - b.fulldate

});

}

/\* CREATE START PAGE IF AVAILABLE

================================================== \*/

if (data.headline != null && data.headline != "" && data.text != null && data.text != "") {

trace("HAS STARTPAGE");

var \_date = {}, td\_num = 0, td;

td = \_dates[0].startdate;

\_date.startdate = new Date(\_dates[0].startdate);

if (td.getMonth() === 0 && td.getDate() == 1 && td.getHours() === 0 && td.getMinutes() === 0 ) {

// trace("YEAR ONLY");

\_date.startdate.setFullYear(td.getFullYear() - 1);

} else if (td.getDate() <= 1 && td.getHours() === 0 && td.getMinutes() === 0) {

// trace("YEAR MONTH");

\_date.startdate.setMonth(td.getMonth() - 1);

} else if (td.getHours() === 0 && td.getMinutes() === 0) {

// trace("YEAR MONTH DAY");

\_date.startdate.setDate(td.getDate() - 1);

} else if (td.getMinutes() === 0) {

// trace("YEAR MONTH DAY HOUR");

\_date.startdate.setHours(td.getHours() - 1);

} else {

// trace("YEAR MONTH DAY HOUR MINUTE");

\_date.startdate.setMinutes(td.getMinutes() - 1);

}

\_date.uniqueid = VMM.Util.unique\_ID(7);

\_date.enddate = \_date.startdate;

\_date.title = data.headline;

\_date.headline = data.headline;

\_date.text = data.text;

\_date.type = "start";

\_date.date = VMM.Date.prettyDate(data.startDate);

\_date.asset = data.asset;

\_date.slug = false;

\_date.needs\_slug = false;

\_date.fulldate = \_date.startdate.getTime();

if (config.embed) {

VMM.fireEvent(global, config.events.headline, \_date.headline);

}

\_dates.unshift(\_date);

}

/\* CUSTOM SORT

================================================== \*/

if (data.type != "storify") {

\_dates.sort(function(a, b){

return a.fulldate - b.fulldate

});

}

onDatesProcessed();

}

};

VMM.Timeline.Config = {};

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Timeline.TimeNav.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* TimeNav

This class handles the bottom timeline navigation.

It requires the VMM.Util class and VMM.Date class

================================================== \*/

if(typeof VMM.Timeline != 'undefined' && typeof VMM.Timeline.TimeNav == 'undefined') {

VMM.Timeline.TimeNav = function(parent, content\_width, content\_height) {

trace("VMM.Timeline.TimeNav");

var events = {},

timespan = {},

layout = parent,

data = [],

era\_markers = [],

markers = [],

interval\_array = [],

interval\_major\_array = [],

tags = [],

current\_marker = 0,

\_active = false,

eras,

content,

timeouts = {

interval\_position: ""

},

timenav\_pos = {

left: "",

visible: {

left: "",

right: ""

}

},

$timenav, $content, $time, $timeintervalminor, $timeinterval, $timeintervalmajor, $timebackground,

$timeintervalbackground, $timenavline, $timenavindicator, $timeintervalminor\_minor, $toolbar, $zoomin, $zoomout, $dragslide;

var timelookup = {day: 24, month: 12, year: 10, hour: 60, minute: 60, second: 1000, decade: 10, century: 100, millenium: 1000, age: 1000000, epoch: 10000000, era: 100000000, eon: 500000000, week: 4.34812141, days\_in\_month: 30.4368499, days\_in\_week: 7, weeks\_in\_month:4.34812141, weeks\_in\_year:52.177457, days\_in\_year: 365.242199, hours\_in\_day: 24 };

var dateFractionBrowser = {day: 86400000, week: 7, month: 30.4166666667, year: 12, hour: 24, minute: 1440, second: 86400, decade: 10, century: 100, millenium: 1000, age: 1000000, epoch: 10000000, era: 100000000, eon: 500000000 };

var interval = {type: "year", number: 10, first: 1970, last: 2011, multiplier: 100, classname:"\_idd", interval\_type:"interval"};

var interval\_major = {type: "year", number: 10, first: 1970, last: 2011, multiplier: 100, classname:"major", interval\_type:"interval major"};

var interval\_macro = {type: "year", number: 10, first: 1970, last: 2011, multiplier: 100, classname:"\_dd\_minor", interval\_type:"interval minor"};

var interval\_calc = {day: {},month: {},year: {},hour: {},minute: {}, second: {},decade: {},century: {},millenium: {},week: {}, age: {}, epoch: {}, era: {}, eon: {} };

/\* ADD to Config

================================================== \*/

var config = VMM.Timeline.Config;

var row\_height = config.nav.marker.height/2;

//config.nav.rows = [1, config.nav.marker.height, config.nav.marker.height\*2];

config.nav.rows = {

full: [1, row\_height\*2, row\_height\*4],

half: [1, row\_height, row\_height\*2, row\_height\*3, row\_height\*4, row\_height\*5],

current: []

}

if (content\_width != null && content\_width != "") {

config.nav.width = content\_width;

}

if (content\_height != null && content\_height != "") {

config.nav.height = content\_height;

}

/\*

config.nav.density = 2;

config.nav.multiplier = {

current: 6,

min: .1,

max: 50

};

\*/

/\* INIT

================================================== \*/

this.init = function(d,e) {

trace('VMM.Timeline.TimeNav init');

// need to evaluate d

// some function to determine type of data and prepare it

if(typeof d != 'undefined') {

this.setData(d, e);

} else {

trace("WAITING ON DATA");

}

};

/\* GETTERS AND SETTERS

================================================== \*/

this.setData = function(d,e) {

if(typeof d != 'undefined') {

data = {};

data = d;

eras = e;

build();

} else{

trace("NO DATA");

}

};

this.setSize = function(w, h) {

if (w != null) {config.width = w};

if (h != null) {config.height = h};

if (\_active) {

reSize();

}

}

this.setMarker = function(n, ease, duration, fast) {

goToMarker(n, ease, duration);

}

this.getCurrentNumber = function() {

return current\_marker;

}

/\* ON EVENT

================================================== \*/

function onConfigSet() {

trace("onConfigSet");

};

function reSize(firstrun) {

config.nav.constraint.left = (config.width/2);

config.nav.constraint.right = config.nav.constraint.right\_min - (config.width/2);

$dragslide.updateConstraint(config.nav.constraint);

VMM.Lib.css($timenavline, "left", Math.round(config.width/2)+2);

VMM.Lib.css($timenavindicator, "left", Math.round(config.width/2)-8);

goToMarker(config.current\_slide, config.ease, config.duration, true, firstrun);

};

function upDate() {

VMM.fireEvent(layout, "UPDATE");

}

function onZoomIn() {

$dragslide.cancelSlide();

if (config.nav.multiplier.current > config.nav.multiplier.min) {

if (config.nav.multiplier.current <= 1) {

config.nav.multiplier.current = config.nav.multiplier.current - .25;

} else {

if (config.nav.multiplier.current > 5) {

if (config.nav.multiplier.current > 16) {

config.nav.multiplier.current = Math.round(config.nav.multiplier.current - 10);

} else {

config.nav.multiplier.current = Math.round(config.nav.multiplier.current - 4);

}

} else {

config.nav.multiplier.current = Math.round(config.nav.multiplier.current - 1);

}

}

if (config.nav.multiplier.current <= 0) {

config.nav.multiplier.current = config.nav.multiplier.min;

}

refreshTimeline();

}

}

function onZoomOut() {

$dragslide.cancelSlide();

if (config.nav.multiplier.current < config.nav.multiplier.max) {

if (config.nav.multiplier.current > 4) {

if (config.nav.multiplier.current > 16) {

config.nav.multiplier.current = Math.round(config.nav.multiplier.current + 10);

} else {

config.nav.multiplier.current = Math.round(config.nav.multiplier.current + 4);

}

} else {

config.nav.multiplier.current = Math.round(config.nav.multiplier.current + 1);

}

if (config.nav.multiplier.current >= config.nav.multiplier.max) {

config.nav.multiplier.current = config.nav.multiplier.max;

}

refreshTimeline();

}

}

function onBackHome(e) {

$dragslide.cancelSlide();

goToMarker(0);

upDate();

}

function onMouseScroll(e) {

var delta = 0,

scroll\_to = 0;

if (!e) {

e = window.event;

}

if (e.originalEvent) {

e = e.originalEvent;

}

if (e.wheelDelta) {

delta = e.wheelDelta/6;

} else if (e.detail) {

delta = -e.detail\*12;

}

if (delta) {

if (e.preventDefault) {

e.preventDefault();

}

e.returnValue = false;

}

// Webkit

if (typeof e.wheelDeltaX != 'undefined' ) {

delta = e.wheelDeltaY/6;

if (Math.abs(e.wheelDeltaX) > Math.abs(e.wheelDeltaY)) {

delta = e.wheelDeltaX/6;

} else {

delta = e.wheelDeltaY/6;

}

}

// Stop from scrolling too far

scroll\_to = VMM.Lib.position($timenav).left + delta;

if (scroll\_to > config.nav.constraint.left) {

scroll\_to = config.width/2;

} else if (scroll\_to < config.nav.constraint.right) {

scroll\_to = config.nav.constraint.right;

}

VMM.Lib.stop($timenav);

//VMM.Lib.animate($timenav, config.duration/2, "linear", {"left": scroll\_to});

VMM.Lib.css($timenav, "left", scroll\_to);

}

var refreshTimeline = function() {

trace("config.nav.multiplier " + config.nav.multiplier.current);

positionMarkers(true);

positionEras(true);

positionInterval($timeinterval, interval\_array, true, true);

positionInterval($timeintervalmajor, interval\_major\_array, true);

config.nav.constraint.left = (config.width/2);

config.nav.constraint.right = config.nav.constraint.right\_min - (config.width/2);

$dragslide.updateConstraint(config.nav.constraint);

};

/\* MARKER EVENTS

================================================== \*/

function onMarkerClick(e) {

$dragslide.cancelSlide();

goToMarker(e.data.number);

upDate();

};

function onMarkerHover(e) {

VMM.Lib.toggleClass(e.data.elem, "zFront");

};

var goToMarker = function(n, ease, duration, fast, firstrun) {

var \_ease = config.ease,

\_duration = config.duration,

is\_last = false,

is\_first = false;

current\_marker = n;

timenav\_pos.left = (config.width/2) - VMM.Lib.position(markers[current\_marker].marker).left;

timenav\_pos.visible.left = Math.abs(timenav\_pos.left) - 100;

timenav\_pos.visible.right = Math.abs(timenav\_pos.left) + config.width + 100;

if (current\_marker == 0) {

is\_first = true;

}

if (current\_marker +1 == markers.length) {

is\_last = true

}

if (ease != null && ease != "") {\_ease = ease};

if (duration != null && duration != "") {\_duration = duration};

// set marker style

for(var i = 0; i < markers.length; i++) {

VMM.Lib.removeClass(markers[i].marker, "active");

}

if (config.start\_page && markers[0].type == "start") {

VMM.Lib.visible(markers[0].marker, false);

VMM.Lib.addClass(markers[0].marker, "start");

}

VMM.Lib.addClass(markers[current\_marker].marker, "active");

// ANIMATE MARKER

VMM.Lib.stop($timenav);

VMM.Lib.animate($timenav, \_duration, \_ease, {"left": timenav\_pos.left});

}

/\* TOUCH EVENTS

================================================== \*/

function onTouchUpdate(e, b) {

VMM.Lib.animate($timenav, b.time/2, config.ease, {"left": b.left});

};

/\* CALCULATIONS

================================================== \*/

var averageMarkerPositionDistance = function() {

var last\_pos = 0,

pos = 0,

pos\_dif = 0,

mp\_diff = [];

for(var i = 0; i < markers.length; i++) {

if (data[i].type == "start") {

} else {

var \_pos = positionOnTimeline(interval, markers[i].relative\_pos),

last\_pos = pos;

pos = \_pos.begin;

pos\_dif = pos - last\_pos;

mp\_diff.push(pos\_dif);

}

}

return VMM.Util.average(mp\_diff).mean;

}

var averageDateDistance = function() {

var last\_dd = 0;

var dd = 0;

var date\_dif = 0;

var date\_diffs = [];

var is\_first\_date = true;

for(var i = 0; i < data.length; i++) {

if (data[i].type == "start") {

trace("DATA DATE IS START")

} else {

var \_dd = data[i].startdate;

last\_dd = dd;

dd = \_dd;

date\_dif = dd - last\_dd;

date\_diffs.push(date\_dif);

}

}

return VMM.Util.average(date\_diffs);

}

var calculateMultiplier = function() {

var temp\_multiplier = config.nav.multiplier.current;

for(var i = 0; i < temp\_multiplier; i++) {

if (averageMarkerPositionDistance() < 75) {

if (config.nav.multiplier.current > 1) {

config.nav.multiplier.current = (config.nav.multiplier.current - 1);

}

}

}

}

var calculateInterval = function() {

// NEED TO REWRITE ALL OF THIS

var \_first = getDateFractions(data[0].startdate);

var \_last = getDateFractions(data[data.length - 1].enddate);

// EON

interval\_calc.eon.type = "eon";

interval\_calc.eon.first = \_first.eons;

interval\_calc.eon.base = Math.floor(\_first.eons);

interval\_calc.eon.last = \_last.eons;

interval\_calc.eon.number = timespan.eons;

interval\_calc.eon.multiplier = timelookup.eons;

interval\_calc.eon.minor = timelookup.eons;

// ERA

interval\_calc.era.type = "era";

interval\_calc.era.first = \_first.eras;

interval\_calc.era.base = Math.floor(\_first.eras);

interval\_calc.era.last = \_last.eras;

interval\_calc.era.number = timespan.eras;

interval\_calc.era.multiplier = timelookup.eras;

interval\_calc.era.minor = timelookup.eras;

// EPOCH

interval\_calc.epoch.type = "epoch";

interval\_calc.epoch.first = \_first.epochs;

interval\_calc.epoch.base = Math.floor(\_first.epochs);

interval\_calc.epoch.last = \_last.epochs;

interval\_calc.epoch.number = timespan.epochs;

interval\_calc.epoch.multiplier = timelookup.epochs;

interval\_calc.epoch.minor = timelookup.epochs;

// AGE

interval\_calc.age.type = "age";

interval\_calc.age.first = \_first.ages;

interval\_calc.age.base = Math.floor(\_first.ages);

interval\_calc.age.last = \_last.ages;

interval\_calc.age.number = timespan.ages;

interval\_calc.age.multiplier = timelookup.ages;

interval\_calc.age.minor = timelookup.ages;

// MILLENIUM

interval\_calc.millenium.type = "millenium";

interval\_calc.millenium.first = \_first.milleniums;

interval\_calc.millenium.base = Math.floor(\_first.milleniums);

interval\_calc.millenium.last = \_last.milleniums;

interval\_calc.millenium.number = timespan.milleniums;

interval\_calc.millenium.multiplier = timelookup.millenium;

interval\_calc.millenium.minor = timelookup.millenium;

// CENTURY

interval\_calc.century.type = "century";

interval\_calc.century.first = \_first.centuries;

interval\_calc.century.base = Math.floor(\_first.centuries);

interval\_calc.century.last = \_last.centuries;

interval\_calc.century.number = timespan.centuries;

interval\_calc.century.multiplier = timelookup.century;

interval\_calc.century.minor = timelookup.century;

// DECADE

interval\_calc.decade.type = "decade";

interval\_calc.decade.first = \_first.decades;

interval\_calc.decade.base = Math.floor(\_first.decades);

interval\_calc.decade.last = \_last.decades;

interval\_calc.decade.number = timespan.decades;

interval\_calc.decade.multiplier = timelookup.decade;

interval\_calc.decade.minor = timelookup.decade;

// YEAR

interval\_calc.year.type = "year";

interval\_calc.year.first = \_first.years;

interval\_calc.year.base = Math.floor(\_first.years);

interval\_calc.year.last = \_last.years;

interval\_calc.year.number = timespan.years;

interval\_calc.year.multiplier = 1;

interval\_calc.year.minor = timelookup.month;

// MONTH

interval\_calc.month.type = "month";

interval\_calc.month.first = \_first.months;

interval\_calc.month.base = Math.floor(\_first.months);

interval\_calc.month.last = \_last.months;

interval\_calc.month.number = timespan.months;

interval\_calc.month.multiplier = 1;

interval\_calc.month.minor = Math.round(timelookup.week);

// WEEK

// NOT DONE

interval\_calc.week.type = "week";

interval\_calc.week.first = \_first.weeks;

interval\_calc.week.base = Math.floor(\_first.weeks);

interval\_calc.week.last = \_last.weeks;

interval\_calc.week.number = timespan.weeks;

interval\_calc.week.multiplier = 1;

interval\_calc.week.minor = 7;

// DAY

interval\_calc.day.type = "day";

interval\_calc.day.first = \_first.days;

interval\_calc.day.base = Math.floor(\_first.days);

interval\_calc.day.last = \_last.days;

interval\_calc.day.number = timespan.days;

interval\_calc.day.multiplier = 1;

interval\_calc.day.minor = 24;

// HOUR

interval\_calc.hour.type = "hour";

interval\_calc.hour.first = \_first.hours;

interval\_calc.hour.base = Math.floor(\_first.hours);

interval\_calc.hour.last = \_last.hours;

interval\_calc.hour.number = timespan.hours;

interval\_calc.hour.multiplier = 1;

interval\_calc.hour.minor = 60;

// MINUTE

interval\_calc.minute.type = "minute";

interval\_calc.minute.first = \_first.minutes;

interval\_calc.minute.base = Math.floor(\_first.minutes);

interval\_calc.minute.last = \_last.minutes;

interval\_calc.minute.number = timespan.minutes;

interval\_calc.minute.multiplier = 1;

interval\_calc.minute.minor = 60;

// SECOND

interval\_calc.second.type = "decade";

interval\_calc.second.first = \_first.seconds;

interval\_calc.second.base = Math.floor(\_first.seconds);

interval\_calc.second.last = \_last.seconds;

interval\_calc.second.number = timespan.seconds;

interval\_calc.second.multiplier = 1;

interval\_calc.second.minor = 10;

}

var getDateFractions = function(the\_date, is\_utc) {

var \_time = {};

\_time.days = the\_date / dateFractionBrowser.day;

\_time.weeks = \_time.days / dateFractionBrowser.week;

\_time.months = \_time.days / dateFractionBrowser.month;

\_time.years = \_time.months / dateFractionBrowser.year;

\_time.hours = \_time.days \* dateFractionBrowser.hour;

\_time.minutes = \_time.days \* dateFractionBrowser.minute;

\_time.seconds = \_time.days \* dateFractionBrowser.second;

\_time.decades = \_time.years / dateFractionBrowser.decade;

\_time.centuries = \_time.years / dateFractionBrowser.century;

\_time.milleniums = \_time.years / dateFractionBrowser.millenium;

\_time.ages = \_time.years / dateFractionBrowser.age;

\_time.epochs = \_time.years / dateFractionBrowser.epoch;

\_time.eras = \_time.years / dateFractionBrowser.era;

\_time.eons = \_time.years / dateFractionBrowser.eon;

/\*

trace("AGES " + \_time.ages);

trace("EPOCHS " + \_time.epochs);

trace("MILLENIUMS " + \_time.milleniums);

trace("CENTURIES " + \_time.centuries);

trace("DECADES " + \_time.decades);

trace("YEARS " + \_time.years);

trace("MONTHS " + \_time.months);

trace("WEEKS " + \_time.weeks);

trace("DAYS " + \_time.days);

trace("HOURS " + \_time.hours);

trace("MINUTES " + \_time.minutes);

trace("SECONDS " + \_time.seconds);

\*/

return \_time;

}

/\* POSITION

Positions elements on the timeline based on date

relative to the calculated interval

================================================== \*/

var positionRelative = function(\_interval, first, last) {

var \_first,

\_last,

\_type = \_interval.type,

timerelative = {start: "", end: "", type: \_type};

/\* FIRST

================================================== \*/

\_first = getDateFractions(first);

timerelative.start = first.months;

if (\_type == "eon") {

timerelative.start = \_first.eons;

} else if (\_type == "era") {

timerelative.start = \_first.eras;

} else if (\_type == "epoch") {

timerelative.start = \_first.epochs;

} else if (\_type == "age") {

timerelative.start = \_first.ages;

} else if (\_type == "millenium") {

timerelative.start = first.milleniums;

} else if (\_type == "century") {

timerelative.start = \_first.centuries;

} else if (\_type == "decade") {

timerelative.start = \_first.decades;

} else if (\_type == "year") {

timerelative.start = \_first.years;

} else if (\_type == "month") {

timerelative.start = \_first.months;

} else if (\_type == "week") {

timerelative.start = \_first.weeks;

} else if (\_type == "day") {

timerelative.start = \_first.days;

} else if (\_type == "hour") {

timerelative.start = \_first.hours;

} else if (\_type == "minute") {

timerelative.start = \_first.minutes;

}

/\* LAST

================================================== \*/

if (type.of(last) == "date") {

\_last = getDateFractions(last);

timerelative.end = last.months;

if (\_type == "eon") {

timerelative.end = \_last.eons;

} else if (\_type == "era") {

timerelative.end = \_last.eras;

} else if (\_type == "epoch") {

timerelative.end = \_last.epochs;

} else if (\_type == "age") {

timerelative.end = \_last.ages;

} else if (\_type == "millenium") {

timerelative.end = last.milleniums;

} else if (\_type == "century") {

timerelative.end = \_last.centuries;

} else if (\_type == "decade") {

timerelative.end = \_last.decades;

} else if (\_type == "year") {

timerelative.end = \_last.years;

} else if (\_type == "month") {

timerelative.end = \_last.months;

} else if (\_type == "week") {

timerelative.end = \_last.weeks;

} else if (\_type == "day") {

timerelative.end = \_last.days;

} else if (\_type == "hour") {

timerelative.end = \_last.hours;

} else if (\_type == "minute") {

timerelative.end = \_last.minutes;

}

} else {

timerelative.end = timerelative.start;

}

return timerelative

}

var positionOnTimeline = function(the\_interval, timerelative) {

return {

begin: (timerelative.start - interval.base) \* (config.nav.interval\_width / config.nav.multiplier.current),

end: (timerelative.end - interval.base) \* (config.nav.interval\_width / config.nav.multiplier.current)

};

}

var positionMarkers = function(is\_animated) {

var row = 2,

previous\_pos = 0,

pos\_offset = -2,

row\_depth = 0,

row\_depth\_sub = 0,

line\_last\_height\_pos = 150,

line\_height = 6,

cur\_mark = 0,

in\_view\_margin = config.width,

pos\_cache\_array = [],

pos\_cache\_max = 6,

in\_view = {

left: timenav\_pos.visible.left - in\_view\_margin,

right: timenav\_pos.visible.right + in\_view\_margin

};

config.nav.minor\_width = config.width;

VMM.Lib.removeClass(".flag", "row1");

VMM.Lib.removeClass(".flag", "row2");

VMM.Lib.removeClass(".flag", "row3");

for(var i = 0; i < markers.length; i++) {

var line,

marker = markers[i],

pos = positionOnTimeline(interval, markers[i].relative\_pos),

row\_pos = 0,

is\_in\_view = false,

pos\_cache\_obj = {id: i, pos: 0, row: 0},

pos\_cache\_close = 0;

// COMPENSATE FOR DATES BEING POITIONED IN THE MIDDLE

pos.begin = Math.round(pos.begin + pos\_offset);

pos.end = Math.round(pos.end + pos\_offset);

line = Math.round(pos.end - pos.begin);

if (current\_marker == i) {

timenav\_pos.left = (config.width/2) - pos;

timenav\_pos.visible.left = Math.abs(timenav\_pos.left);

timenav\_pos.visible.right = Math.abs(timenav\_pos.left) + config.width;

in\_view.left = timenav\_pos.visible.left - in\_view\_margin;

in\_view.right = timenav\_pos.visible.right + in\_view\_margin;

}

if (Math.abs(pos.begin) >= in\_view.left && Math.abs(pos.begin) <= in\_view.right ) {

is\_in\_view = true;

}

// APPLY POSITION TO MARKER

if (is\_animated) {

VMM.Lib.stop(marker.marker);

VMM.Lib.animate(marker.marker, config.duration/2, config.ease, {"left": pos.begin});

} else {

VMM.Lib.stop(marker.marker);

VMM.Lib.css(marker.marker, "left", pos.begin);

}

if (i == current\_marker) {

cur\_mark = pos.begin;

}

// EVENT LENGTH LINE

if (line > 5) {

VMM.Lib.css(marker.lineevent, "height", line\_height);

VMM.Lib.css(marker.lineevent, "top", line\_last\_height\_pos);

if (is\_animated) {

VMM.Lib.animate(marker.lineevent, config.duration/2, config.ease, {"width": line});

} else {

VMM.Lib.css(marker.lineevent, "width", line);

}

}

// CONTROL ROW POSITION

if (tags.length > 0) {

for (var k = 0; k < tags.length; k++) {

if (k < config.nav.rows.current.length) {

if (marker.tag == tags[k]) {

row = k;

if (k == config.nav.rows.current.length - 1) {

trace("ON LAST ROW");

VMM.Lib.addClass(marker.flag, "flag-small-last");

}

}

}

}

row\_pos = config.nav.rows.current[row];

} else {

/\*

pos\_cache\_close = 0;

for (var l = 0; l < pos\_cache\_array.length; l++) {

if (pos.begin - pos\_cache\_array[l].pos.begin < ((config.nav.marker.width/2))) {

pos\_cache\_close++;

trace("POS CACHE TOO CLOSE");

markers[pos\_cache\_array[l].id].full = false;

VMM.Lib.addClass(markers[pos\_cache\_array[l].id].flag, "flag-small");

}

}

if (pos\_cache\_close > 3) {

trace("POS CACHE TOO CLOSE GRATER THAN 2");

marker.full = false;

VMM.Lib.addClass(marker.flag, "flag-small");

if (row < config.nav.rows.current.length - 1) {

row ++;

} else {

row = 0;

row\_depth ++;

}

if (config.nav.rows.current == config.nav.rows.full) {

if (row\_depth\_sub == 0) {

row\_depth\_sub = 1;

} else {

row\_depth\_sub = 0;

}

row\_pos = config.nav.rows.half[ (row\*2) + row\_depth\_sub ];

} else {

row\_pos = config.nav.rows.half[row];

}

} else {

if (!marker.full) {

VMM.Lib.removeClass(markers[i].flag, "flag-small");

marker.full = true;

}

if (pos.begin - previous\_pos.begin < (config.nav.marker.width + config.spacing)) {

if (row < config.nav.rows.full.length - 1) {

row ++;

} else {

row = 0;

row\_depth ++;

}

} else {

row\_depth = 1;

row = 1;

}

row\_pos = config.nav.rows.full[row];

}

\*/

if (pos.begin - previous\_pos.begin < (config.nav.marker.width + config.spacing)) {

if (row < config.nav.rows.current.length - 1) {

row ++;

} else {

row = 0;

row\_depth ++;

}

} else {

row\_depth = 1;

row = 1;

}

row\_pos = config.nav.rows.current[row];

/\*

if (row\_depth > 2) {

marker.full = false;

VMM.Lib.addClass(marker.flag, "flag-small");

trace(config.nav.rows.current);

trace(config.nav.rows.full);

if (row\_depth\_sub == 0) {

row\_depth\_sub = 1;

} else {

row\_depth\_sub = 0;

}

if (config.nav.rows.current == config.nav.rows.full) {

trace(config.nav.rows.half[(row\*2) + 1]);

trace(config.nav.rows.full[row]);

row\_pos = config.nav.rows.half[ (row\*2) + row\_depth\_sub ];

}

}

\*/

}

// SET LAST MARKER POSITION

previous\_pos = pos;

// POSITION CACHE

pos\_cache\_obj.pos = pos;

pos\_cache\_obj.row = row;

pos\_cache\_array.push(pos\_cache\_obj);

if (pos\_cache\_array.length > pos\_cache\_max) {

pos\_cache\_array.remove(0);

}

//if (is\_animated && is\_in\_view) {

if (is\_animated) {

VMM.Lib.stop(marker.flag);

VMM.Lib.animate(marker.flag, config.duration, config.ease, {"top": row\_pos});

} else {

VMM.Lib.stop(marker.flag);

VMM.Lib.css(marker.flag, "top", row\_pos);

}

// IS THE MARKER A REPRESENTATION OF A START SCREEN?

if (config.start\_page && markers[i].type == "start") {

VMM.Lib.visible(marker.marker, false);

}

if (pos > config.nav.minor\_width) {

config.nav.minor\_width = pos;

}

if (pos < config.nav.minor\_left) {

config.nav.minor\_left = pos;

}

}

// ANIMATE THE TIMELINE TO ADJUST TO CHANGES

VMM.Lib.stop($timenav);

VMM.Lib.animate($timenav, config.duration/2, config.ease, {"left": (config.width/2) - (cur\_mark)});

//VMM.Lib.delay\_animate(config.duration, $timenav, config.duration/2, config.ease, {"left": (config.width/2) - (cur\_mark)});

}

var positionEras = function(is\_animated) {

for(var i = 0; i < era\_markers.length; i++) {

var era = era\_markers[i],

pos = positionOnTimeline(interval, era.relative\_pos),

row\_pos = 0,

era\_height = config.nav.marker.height \* config.nav.rows.full.length,

era\_length = pos.end - pos.begin;

// CONTROL ROW POSITION

if (era.tag != "") {

era\_height = (config.nav.marker.height \* config.nav.rows.full.length) / config.nav.rows.current.length;

for (var p = 0; p < tags.length; p++) {

if (p < config.nav.rows.current.length) {

if (era.tag == tags[p]) {

row = p;

}

}

}

row\_pos = config.nav.rows.current[row];

} else {

row\_pos = -1;

}

// APPLY POSITION TO MARKER

if (is\_animated) {

VMM.Lib.stop(era.content);

VMM.Lib.stop(era.text\_content);

VMM.Lib.animate(era.content, config.duration/2, config.ease, {"top": row\_pos, "left": pos.begin, "width": era\_length, "height":era\_height});

VMM.Lib.animate(era.text\_content, config.duration/2, config.ease, {"left": pos.begin});

} else {

VMM.Lib.stop(era.content);

VMM.Lib.stop(era.text\_content);

VMM.Lib.css(era.content, "left", pos.begin);

VMM.Lib.css(era.content, "width", era\_length);

VMM.Lib.css(era.content, "height", era\_height);

VMM.Lib.css(era.content, "top", row\_pos);

VMM.Lib.css(era.text\_content, "left", pos.begin);

}

}

}

var positionInterval = function(the\_main\_element, the\_intervals, is\_animated, is\_minor) {

var last\_position = 0,

last\_position\_major = 0,

//in\_view\_margin = (config.nav.minor\_width/config.nav.multiplier.current)/2,

in\_view\_margin = config.width,

in\_view = {

left: timenav\_pos.visible.left - in\_view\_margin,

right: timenav\_pos.visible.right + in\_view\_margin

}

not\_too\_many = true;

config.nav.minor\_left = 0;

if (the\_intervals.length > 100) {

not\_too\_many = false;

trace("TOO MANY " + the\_intervals.length);

}

for(var i = 0; i < the\_intervals.length; i++) {

var \_interval = the\_intervals[i].element,

\_interval\_date = the\_intervals[i].date,

\_interval\_visible = the\_intervals[i].visible,

\_pos = positionOnTimeline(interval, the\_intervals[i].relative\_pos),

pos = \_pos.begin,

\_animation = the\_intervals[i].animation,

is\_visible = true,

is\_in\_view = false,

pos\_offset = 50;

\_animation.pos = pos;

\_animation.animate = false;

if (Math.abs(pos) >= in\_view.left && Math.abs(pos) <= in\_view.right ) {

is\_in\_view = true;

}

if (true) {

// CONDENSE WHAT IS DISPLAYED

if (config.nav.multiplier.current > 16 && is\_minor) {

is\_visible = false;

} else {

if ((pos - last\_position) < 65 ) {

if ((pos - last\_position) < 35 ) {

if (i%4 == 0) {

if (pos == 0) {

is\_visible = false;

}

} else {

is\_visible = false;

}

} else {

if (!VMM.Util.isEven(i)) {

is\_visible = false;

}

}

}

}

if (is\_visible) {

if (the\_intervals[i].is\_detached) {

VMM.Lib.append(the\_main\_element, \_interval);

the\_intervals[i].is\_detached = false;

}

} else {

the\_intervals[i].is\_detached = true;

VMM.Lib.detach(\_interval);

}

if (\_interval\_visible) {

if (!is\_visible) {

\_animation.opacity = "0";

if (is\_animated && not\_too\_many) {

\_animation.animate = true;

}

the\_intervals[i].interval\_visible = false;

} else {

\_animation.opacity = "100";

if (is\_animated && is\_in\_view) {

\_animation.animate = true;

}

}

} else {

\_animation.opacity = "100";

if (is\_visible) {

if (is\_animated && not\_too\_many) {

\_animation.animate = true;

} else {

if (is\_animated && is\_in\_view) {

\_animation.animate = true;

}

}

the\_intervals[i].interval\_visible = true;

} else {

if (is\_animated && not\_too\_many) {

\_animation.animate = true;

}

}

}

last\_position = pos;

if (pos > config.nav.minor\_width) {

config.nav.minor\_width = pos;

//config.nav.constraint.right\_min = -pos;

//config.nav.constraint.right = config.nav.constraint.right\_min + (config.width/2);

}

if (pos < config.nav.minor\_left) {

config.nav.minor\_left = pos;

//config.nav.constraint.right\_min = pos;

//config.nav.constraint.right = config.nav.constraint.right\_min + (config.width/2);

}

}

if (\_animation.animate) {

VMM.Lib.animate(\_interval, config.duration/2, config.ease, {opacity: \_animation.opacity, left: \_animation.pos});

} else {

VMM.Lib.css(\_interval, "opacity", \_animation.opacity);

VMM.Lib.css(\_interval, "left", pos);

}

}

config.nav.constraint.right\_min = -(config.nav.minor\_width)+(config.width);

config.nav.constraint.right = config.nav.constraint.right\_min + (config.width/2);

VMM.Lib.css($timeintervalminor\_minor, "left", config.nav.minor\_left - (config.width)/2);

VMM.Lib.width($timeintervalminor\_minor, (config.nav.minor\_width)+(config.width) + Math.abs(config.nav.minor\_left) );

//trace((config.nav.minor\_width/config.nav.multiplier.current)/2)

/\*

for(var k = 0; k < the\_intervals.length; k++) {

var \_animation = the\_intervals[k].animation;

if (\_animation.animate) {

var \_interval = the\_intervals[k].interval\_element;

VMM.Lib.animate(\_interval, config.duration/2, config.ease, {opacity: \_animation.opacity, left: \_animation.pos}, "interval\_que");

}

}

\*/

}

/\* Interval Elements

================================================== \*/

var createIntervalElements = function(\_interval, \_array, \_element\_parent) {

var inc\_time = 0,

\_first\_run = true,

\_last\_pos = 0,

\_largest\_pos = 0,

\_timezone\_offset,

\_first\_date,

\_last\_date,

int\_number = Math.ceil(\_interval.number) + 2,

firefox = {

flag: false,

offset: 0

};

VMM.attachElement(\_element\_parent, "");

\_interval.date = new Date(data[0].startdate.getFullYear(), 0, 1, 0,0,0);

\_timezone\_offset = \_interval.date.getTimezoneOffset();

for(var i = 0; i < int\_number; i++) {

trace(\_interval.type);

var \_is\_year = false,

int\_obj = {

element: VMM.appendAndGetElement(\_element\_parent, "<div>", \_interval.classname),

date: new Date(data[0].startdate.getFullYear(), 0, 1, 0,0,0),

visible: false,

date\_string: "",

type: \_interval.interval\_type,

relative\_pos: 0,

is\_detached: false,

animation: {

animate: false,

pos: "",

opacity: "100"

}

};

if (\_interval.type == "eon") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 500000000) \* 500000000;

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 500000000));

\_is\_year = true;

} else if (\_interval.type == "era") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 100000000) \* 100000000;

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 100000000));

\_is\_year = true;

} else if (\_interval.type == "epoch") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 10000000) \* 10000000

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 10000000));

\_is\_year = true;

} else if (\_interval.type == "age") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 1000000) \* 1000000

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 1000000));

\_is\_year = true;

} else if (\_interval.type == "millenium") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 1000) \* 1000;

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 1000));

\_is\_year = true;

} else if (\_interval.type == "century") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 100) \* 100

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 100));

\_is\_year = true;

} else if (\_interval.type == "decade") {

if (\_first\_run) {

\_first\_date = Math.floor(data[0].startdate.getFullYear() / 10) \* 10;

}

int\_obj.date.setFullYear(\_first\_date + (inc\_time \* 10));

\_is\_year = true;

} else if (\_interval.type == "year") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getFullYear();

}

int\_obj.date.setFullYear(\_first\_date + inc\_time);

\_is\_year = true;

} else if (\_interval.type == "month") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getMonth();

}

int\_obj.date.setMonth(\_first\_date + inc\_time);

} else if (\_interval.type == "week") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getMonth();

}

int\_obj.date.setMonth(data[0].startdate.getMonth());

int\_obj.date.setDate(\_first\_date + (inc\_time \* 7) );

} else if (\_interval.type == "day") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getDate();

}

int\_obj.date.setMonth(data[0].startdate.getMonth());

int\_obj.date.setDate(\_first\_date + inc\_time);

} else if (\_interval.type == "hour") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getHours();

}

int\_obj.date.setMonth(data[0].startdate.getMonth());

int\_obj.date.setDate(data[0].startdate.getDate());

int\_obj.date.setHours(\_first\_date + inc\_time);

} else if (\_interval.type == "minute") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getMinutes();

}

int\_obj.date.setMonth(data[0].startdate.getMonth());

int\_obj.date.setDate(data[0].startdate.getDate());

int\_obj.date.setHours(data[0].startdate.getHours());

int\_obj.date.setMinutes(\_first\_date + inc\_time);

} else if (\_interval.type == "second") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getSeconds();

}

int\_obj.date.setMonth(data[0].startdate.getMonth());

int\_obj.date.setDate(data[0].startdate.getDate());

int\_obj.date.setHours(data[0].startdate.getHours());

int\_obj.date.setMinutes(data[0].startdate.getMinutes());

int\_obj.date.setSeconds(\_first\_date + inc\_time);

} else if (\_interval.type == "millisecond") {

if (\_first\_run) {

\_first\_date = data[0].startdate.getMilliseconds();

}

int\_obj.date.setMonth(data[0].startdate.getMonth());

int\_obj.date.setDate(data[0].startdate.getDate());

int\_obj.date.setHours(data[0].startdate.getHours());

int\_obj.date.setMinutes(data[0].startdate.getMinutes());

int\_obj.date.setSeconds(data[0].startdate.getSeconds());

int\_obj.date.setMilliseconds(\_first\_date + inc\_time);

}

// FIX WEIRD FIREFOX BUG FOR GMT TIME FORMATTING

if (VMM.Browser.browser == "Firefox") {

if (int\_obj.date.getFullYear() == "1970" && int\_obj.date.getTimezoneOffset() != \_timezone\_offset) {

trace("FIREFOX 1970 TIMEZONE OFFSET " + int\_obj.date.getTimezoneOffset() + " SHOULD BE " + \_timezone\_offset);

trace(\_interval.type + " " + \_interval.date);

// try and fix firefox bug, if not the flag will catch it

firefox.offset = (int\_obj.date.getTimezoneOffset()/60);

firefox.flag = true;

int\_obj.date.setHours(int\_obj.date.getHours() + firefox.offset );

} else if (firefox.flag) {

// catch the bug the second time around

firefox.flag = false;

int\_obj.date.setHours(int\_obj.date.getHours() + firefox.offset );

if (\_is\_year) {

firefox.flag = true;

}

}

}

if (\_is\_year) {

if ( int\_obj.date.getFullYear() < 0 ) {

int\_obj.date\_string = Math.abs( int\_obj.date.getFullYear() ).toString() + " B.C.";

} else {

int\_obj.date\_string = int\_obj.date.getFullYear();

}

} else {

int\_obj.date\_string = VMM.Date.prettyDate(int\_obj.date, true);

}

// Increment Time

inc\_time = inc\_time + 1;

// No longer first run

\_first\_run = false;

int\_obj.relative\_pos = positionRelative(interval, int\_obj.date);

\_last\_pos = int\_obj.relative\_pos.begin;

if (int\_obj.relative\_pos.begin > \_largest\_pos) {

\_largest\_pos = int\_obj.relative\_pos.begin;

}

// Add the time string to the element and position it.

VMM.appendElement(int\_obj.element, int\_obj.date\_string);

VMM.Lib.css(int\_obj.element, "text-indent", -(VMM.Lib.width(int\_obj.element)/2));

VMM.Lib.css(int\_obj.element, "opacity", "0");

// add the interval element to the array

\_array.push(int\_obj);

}

VMM.Lib.width($timeintervalminor\_minor, \_largest\_pos);

positionInterval(\_element\_parent, \_array);

}

/\* BUILD

================================================== \*/

var build = function() {

VMM.attachElement(layout, "");

$timenav = VMM.appendAndGetElement(layout, "<div>", "timenav");

$content = VMM.appendAndGetElement($timenav, "<div>", "content");

$time = VMM.appendAndGetElement($timenav, "<div>", "time");

$timeintervalminor = VMM.appendAndGetElement($time, "<div>", "time-interval-minor");

$timeintervalminor\_minor = VMM.appendAndGetElement($timeintervalminor, "<div>", "minor");

$timeintervalmajor = VMM.appendAndGetElement($time, "<div>", "time-interval-major");

$timeinterval = VMM.appendAndGetElement($time, "<div>", "time-interval");

$timebackground = VMM.appendAndGetElement(layout, "<div>", "timenav-background");

$timenavline = VMM.appendAndGetElement($timebackground, "<div>", "timenav-line");

$timenavindicator = VMM.appendAndGetElement($timebackground, "<div>", "timenav-indicator");

$timeintervalbackground = VMM.appendAndGetElement($timebackground, "<div>", "timenav-interval-background", "<div class='top-highlight'></div>");

$toolbar = VMM.appendAndGetElement(layout, "<div>", "toolbar");

buildInterval();

buildMarkers();

buildEras();

calculateMultiplier();

positionMarkers();

positionEras();

positionInterval($timeinterval, interval\_array, false, true);

positionInterval($timeintervalmajor, interval\_major\_array);

if (config.start\_page) {

$backhome = VMM.appendAndGetElement($toolbar, "<div>", "back-home", "<div class='icon'></div>");

VMM.bindEvent(".back-home", onBackHome, "click");

VMM.Lib.css($toolbar, "top", 27);

VMM.Lib.attribute($backhome, "title", VMM.master\_config.language.messages.return\_to\_title);

VMM.Lib.attribute($backhome, "rel", "tooltip");

}

$zoomin = VMM.appendAndGetElement($toolbar, "<div>", "zoom-in", "<div class='icon'></div>");

$zoomout = VMM.appendAndGetElement($toolbar, "<div>", "zoom-out", "<div class='icon'></div>");

// MAKE TIMELINE DRAGGABLE/TOUCHABLE

$dragslide = new VMM.DragSlider;

$dragslide.createPanel(layout, $timenav, config.nav.constraint, config.touch);

// ZOOM EVENTS

VMM.bindEvent($zoomin, onZoomIn, "click");

VMM.bindEvent($zoomout, onZoomOut, "click");

if (!config.touch) {

// TOOLTIP

VMM.Lib.attribute($zoomin, "title", VMM.master\_config.language.messages.expand\_timeline);

VMM.Lib.attribute($zoomin, "rel", "tooltip");

VMM.Lib.attribute($zoomout, "title", VMM.master\_config.language.messages.contract\_timeline);

VMM.Lib.attribute($zoomout, "rel", "tooltip");

$toolbar.tooltip({selector: "div[rel=tooltip]", placement: "right"});

// MOUSE EVENTS

VMM.bindEvent(layout, onMouseScroll, 'DOMMouseScroll');

VMM.bindEvent(layout, onMouseScroll, 'mousewheel');

}

VMM.fireEvent(layout, "LOADED");

\_active = true;

reSize(true);

// USER CONFIGURABLE ADJUSTMENT TO DEFAULT ZOOM

if (config.nav.zoom.adjust != 0) {

if (config.nav.zoom.adjust < 0) {

for(var i = 0; i < Math.abs(config.nav.zoom.adjust); i++) {

onZoomOut();

}

} else {

for(var j = 0; j < config.nav.zoom.adjust; j++) {

onZoomIn();

}

}

}

};

var buildInterval = function() {

// CALCULATE INTERVAL

timespan = getDateFractions((data[data.length - 1].enddate) - (data[0].startdate), true);

trace(timespan);

calculateInterval();

/\* DETERMINE DEFAULT INTERVAL TYPE

millenium, ages, epoch, era and eon are not working yet

================================================== \*/

/\*

if (timespan.eons > data.length / config.nav.density) {

interval = interval\_calc.eon;

interval\_major = interval\_calc.eon;

interval\_macro = interval\_calc.era;

} else if (timespan.eras > data.length / config.nav.density) {

interval = interval\_calc.era;

interval\_major = interval\_calc.eon;

interval\_macro = interval\_calc.epoch;

} else if (timespan.epochs > data.length / config.nav.density) {

interval = interval\_calc.epoch;

interval\_major = interval\_calc.era;

interval\_macro = interval\_calc.age;

} else if (timespan.ages > data.length / config.nav.density) {

interval = interval\_calc.ages;

interval\_major = interval\_calc.epoch;

interval\_macro = interval\_calc.millenium;

} else if (timespan.milleniums > data.length / config.nav.density) {

interval = interval\_calc.millenium;

interval\_major = interval\_calc.age;

interval\_macro = interval\_calc.century;

} else

\*/

if (timespan.centuries > data.length / config.nav.density) {

interval = interval\_calc.century;

interval\_major = interval\_calc.millenium;

interval\_macro = interval\_calc.decade;

} else if (timespan.decades > data.length / config.nav.density) {

interval = interval\_calc.decade;

interval\_major = interval\_calc.century;

interval\_macro = interval\_calc.year;

} else if (timespan.years > data.length / config.nav.density) {

interval = interval\_calc.year;

interval\_major = interval\_calc.decade;

interval\_macro = interval\_calc.month;

} else if (timespan.months > data.length / config.nav.density) {

interval = interval\_calc.month;

interval\_major = interval\_calc.year;

interval\_macro = interval\_calc.day;

} else if (timespan.days > data.length / config.nav.density) {

interval = interval\_calc.day;

interval\_major = interval\_calc.month;

interval\_macro = interval\_calc.hour;

} else if (timespan.hours > data.length / config.nav.density) {

interval = interval\_calc.hour;

interval\_major = interval\_calc.day;

interval\_macro = interval\_calc.minute;

} else if (timespan.minutes > data.length / config.nav.density) {

interval = interval\_calc.minute;

interval\_major = interval\_calc.hour;

interval\_macro = interval\_calc.second;

} else if (timespan.seconds > data.length / config.nav.density) {

interval = interval\_calc.second;

interval\_major = interval\_calc.minute;

interval\_macro = interval\_calc.second;

} else {

trace("NO IDEA WHAT THE TYPE SHOULD BE");

interval = interval\_calc.day;

interval\_major = interval\_calc.month;

interval\_macro = interval\_calc.hour;

}

trace("INTERVAL TYPE: " + interval.type);

trace("INTERVAL MAJOR TYPE: " + interval\_major.type);

createIntervalElements(interval, interval\_array, $timeinterval);

createIntervalElements(interval\_major, interval\_major\_array, $timeintervalmajor);

// Cleanup duplicate interval elements between normal and major

for(var i = 0; i < interval\_array.length; i++) {

for(var j = 0; j < interval\_major\_array.length; j++) {

if (interval\_array[i].date\_string == interval\_major\_array[j].date\_string) {

VMM.attachElement(interval\_array[i].element, "");

}

}

}

}

var buildMarkers = function() {

var row = 2,

lpos = 0,

row\_depth = 0;

markers = [];

era\_markers = [];

for(var i = 0; i < data.length; i++) {

var \_marker, \_marker\_flag, \_marker\_content, \_marker\_dot, \_marker\_line, \_marker\_line\_event, \_marker\_title = "", has\_title = false;

\_marker = VMM.appendAndGetElement($content, "<div>", "marker");

\_marker\_flag = VMM.appendAndGetElement(\_marker, "<div>", "flag");

\_marker\_content = VMM.appendAndGetElement(\_marker\_flag, "<div>", "flag-content");

\_marker\_dot = VMM.appendAndGetElement(\_marker, "<div>", "dot");

\_marker\_line = VMM.appendAndGetElement(\_marker, "<div>", "line");

\_marker\_line\_event = VMM.appendAndGetElement(\_marker\_line, "<div>", "event-line");

\_marker\_relative\_pos = positionRelative(interval, data[i].startdate, data[i].enddate);

\_marker\_thumb = "";

// THUMBNAIL

if (data[i].asset != null && data[i].asset != "") {

VMM.appendElement(\_marker\_content, VMM.MediaElement.thumbnail(data[i].asset, 24, 24, data[i].uniqueid));

} else {

VMM.appendElement(\_marker\_content, "<div style='margin-right:7px;height:50px;width:2px;float:left;'></div>");

}

// ADD DATE AND TITLE

if (data[i].title == "" || data[i].title == " " ) {

trace("TITLE NOTHING")

if (typeof data[i].slug != 'undefined' && data[i].slug != "") {

trace("SLUG")

\_marker\_title = VMM.Util.untagify(data[i].slug);

has\_title = true;

} else {

var m = VMM.MediaType(data[i].asset.media);

if (m.type == "quote" || m.type == "unknown") {

\_marker\_title = VMM.Util.untagify(m.id);

has\_title = true;

} else if (m.type == "twitter") {

has\_title = false;

VMM.appendElement(\_marker\_content, "<h3 id='text\_thumb\_" + m.id + "'>" + \_marker\_title + "</h3>");

} else {

has\_title = false;

}

}

} else if (data[i].title != "" || data[i].title != " ") {

trace(data[i].title)

\_marker\_title = VMM.Util.untagify(data[i].title);

has\_title = true;

} else {

trace("TITLE SLUG NOT FOUND " + data[i].slug)

}

if (has\_title) {

VMM.appendElement(\_marker\_content, "<h3>" + \_marker\_title + "</h3>");

}

// ADD ID

VMM.Lib.attr(\_marker, "id", ( "marker\_" + data[i].uniqueid).toString() );

// MARKER CLICK

VMM.bindEvent(\_marker\_flag, onMarkerClick, "", {number: i});

VMM.bindEvent(\_marker\_flag, onMarkerHover, "mouseenter mouseleave", {number: i, elem:\_marker\_flag});

var \_marker\_obj = {

marker: \_marker,

flag: \_marker\_flag,

lineevent: \_marker\_line\_event,

type: "marker",

full: true,

relative\_pos: \_marker\_relative\_pos,

tag: data[i].tag

};

if (data[i].type == "start") {

trace("BUILD MARKER HAS START PAGE");

config.start\_page = true;

\_marker\_obj.type = "start";

}

if (data[i].type == "storify") {

\_marker\_obj.type = "storify";

}

if (data[i].tag) {

tags.push(data[i].tag);

}

markers.push(\_marker\_obj);

}

// CREATE TAGS

tags = VMM.Util.deDupeArray(tags);

if (tags.length > 2) {

config.nav.rows.current = config.nav.rows.half;

} else {

config.nav.rows.current = config.nav.rows.full;

}

for(var k = 0; k < tags.length; k++) {

if (k < config.nav.rows.current.length) {

var tag\_element = VMM.appendAndGetElement($timebackground, "<div>", "timenav-tag");

VMM.Lib.addClass(tag\_element, "timenav-tag-row-" + (k+1));

if (tags.length > 2) {

VMM.Lib.addClass(tag\_element, "timenav-tag-size-half");

} else {

VMM.Lib.addClass(tag\_element, "timenav-tag-size-full");

}

VMM.appendElement(tag\_element, "<div><h3>" + tags[k] + "</h3></div>");

}

}

// RESIZE FLAGS IF NEEDED

if (tags.length > 2) {

for(var l = 0; l < markers.length; l++) {

VMM.Lib.addClass(markers[l].flag, "flag-small");

markers[l].full = false;

}

}

}

var buildEras = function() {

var number\_of\_colors = 6,

current\_color = 0;

// CREATE ERAS

for(var j = 0; j < eras.length; j++) {

var era = {

content: VMM.appendAndGetElement($content, "<div>", "era"),

text\_content: VMM.appendAndGetElement($timeinterval, "<div>", "era"),

startdate: VMM.Date.parse(eras[j].startDate),

enddate: VMM.Date.parse(eras[j].endDate),

title: eras[j].headline,

uniqueid: VMM.Util.unique\_ID(6),

tag: "",

relative\_pos: ""

},

st = VMM.Date.prettyDate(era.startdate),

en = VMM.Date.prettyDate(era.enddate),

era\_text = "";

if (typeof eras[j].tag != "undefined") {

era.tag = eras[j].tag;

}

era.relative\_pos = positionRelative(interval, era.startdate, era.enddate);

VMM.Lib.attr(era.content, "id", era.uniqueid);

VMM.Lib.attr(era.text\_content, "id", era.uniqueid + "\_text");

//VMM.Lib.css(era.content, "background", era.color);

era\_text += "<div>&nbsp;";

//era\_text += "<h3>" + VMM.Util.unlinkify(era.title) + "</h3>"

if (st != en) {

//era\_text += "<h4>" + st + " &mdash; " + en + "</h4>";

} else {

//era\_text += "<h4>" + st + "</h4>";

}

era\_text += "</div>";

// Background Color

VMM.Lib.addClass(era.content, "era"+(current\_color+1));

VMM.Lib.addClass(era.text\_content, "era"+(current\_color+1));

if (current\_color < number\_of\_colors) {

current\_color++;

} else {

current\_color = 0;

}

VMM.appendElement(era.content, era\_text);

VMM.appendElement(era.text\_content, VMM.Util.unlinkify(era.title));

era\_markers.push(era);

}

}

};

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Begin VMM.Timeline.DataObj.js

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* TIMELINE SOURCE DATA PROCESSOR

================================================== \*/

if (typeof VMM.Timeline !== 'undefined' && typeof VMM.Timeline.DataObj == 'undefined') {

VMM.Timeline.DataObj = {

data\_obj: {},

model\_array: [],

getData: function (raw\_data) {

VMM.Timeline.DataObj.data\_obj = {};

VMM.fireEvent(global, VMM.Timeline.Config.events.messege, VMM.Timeline.Config.language.messages.loading\_timeline);

if (type.of(raw\_data) == "object") {

trace("DATA SOURCE: JSON OBJECT");

VMM.Timeline.DataObj.parseJSON(raw\_data);

} else if (type.of(raw\_data) == "string") {

if (raw\_data.match("%23")) {

trace("DATA SOURCE: TWITTER SEARCH");

VMM.Timeline.DataObj.model.tweets.getData("%23medill");

} else if ( raw\_data.match("spreadsheet") ) {

trace("DATA SOURCE: GOOGLE SPREADSHEET");

VMM.Timeline.DataObj.model.googlespreadsheet.getData(raw\_data);

} else if (raw\_data.match("storify.com")) {

trace("DATA SOURCE: STORIFY");

VMM.Timeline.DataObj.model.storify.getData(raw\_data);

//http://api.storify.com/v1/stories/number10gov/g8-and-nato-chicago-summit

} else {

trace("DATA SOURCE: JSON");

VMM.getJSON(raw\_data, VMM.Timeline.DataObj.parseJSON);

}

} else if (type.of(raw\_data) == "html") {

trace("DATA SOURCE: HTML");

VMM.Timeline.DataObj.parseHTML(raw\_data);

} else {

trace("DATA SOURCE: UNKNOWN");

}

},

parseHTML: function (d) {

trace("parseHTML");

trace("WARNING: THIS IS STILL ALPHA AND WILL NOT WORK WITH ID's other than #timeline");

var \_data\_obj = VMM.Timeline.DataObj.data\_template\_obj;

/\* Timeline start slide

================================================== \*/

if (VMM.Lib.find("#timeline section", "time")[0]) {

\_data\_obj.timeline.startDate = VMM.Lib.html(VMM.Lib.find("#timeline section", "time")[0]);

\_data\_obj.timeline.headline = VMM.Lib.html(VMM.Lib.find("#timeline section", "h2"));

\_data\_obj.timeline.text = VMM.Lib.html(VMM.Lib.find("#timeline section", "article"));

var found\_main\_media = false;

if (VMM.Lib.find("#timeline section", "figure img").length != 0) {

found\_main\_media = true;

\_data\_obj.timeline.asset.media = VMM.Lib.attr(VMM.Lib.find("#timeline section", "figure img"), "src");

} else if (VMM.Lib.find("#timeline section", "figure a").length != 0) {

found\_main\_media = true;

\_data\_obj.timeline.asset.media = VMM.Lib.attr(VMM.Lib.find("#timeline section", "figure a"), "href");

} else {

//trace("NOT FOUND");

}

if (found\_main\_media) {

if (VMM.Lib.find("#timeline section", "cite").length != 0) {

\_data\_obj.timeline.asset.credit = VMM.Lib.html(VMM.Lib.find("#timeline section", "cite"));

}

if (VMM.Lib.find(this, "figcaption").length != 0) {

\_data\_obj.timeline.asset.caption = VMM.Lib.html(VMM.Lib.find("#timeline section", "figcaption"));

}

}

}

/\* Timeline Date Slides

================================================== \*/

VMM.Lib.each("#timeline li", function(i, elem){

var valid\_date = false;

var \_date = {

"type":"default",

"startDate":"",

"headline":"",

"text":"",

"asset":

{

"media":"",

"credit":"",

"caption":""

},

"tags":"Optional"

};

if (VMM.Lib.find(this, "time") != 0) {

valid\_date = true;

\_date.startDate = VMM.Lib.html(VMM.Lib.find(this, "time")[0]);

if (VMM.Lib.find(this, "time")[1]) {

\_date.endDate = VMM.Lib.html(VMM.Lib.find(this, "time")[1]);

}

\_date.headline = VMM.Lib.html(VMM.Lib.find(this, "h3"));

\_date.text = VMM.Lib.html(VMM.Lib.find(this, "article"));

var found\_media = false;

if (VMM.Lib.find(this, "figure img").length != 0) {

found\_media = true;

\_date.asset.media = VMM.Lib.attr(VMM.Lib.find(this, "figure img"), "src");

} else if (VMM.Lib.find(this, "figure a").length != 0) {

found\_media = true;

\_date.asset.media = VMM.Lib.attr(VMM.Lib.find(this, "figure a"), "href");

} else {

//trace("NOT FOUND");

}

if (found\_media) {

if (VMM.Lib.find(this, "cite").length != 0) {

\_date.asset.credit = VMM.Lib.html(VMM.Lib.find(this, "cite"));

}

if (VMM.Lib.find(this, "figcaption").length != 0) {

\_date.asset.caption = VMM.Lib.html(VMM.Lib.find(this, "figcaption"));

}

}

trace(\_date);

\_data\_obj.timeline.date.push(\_date);

}

});

VMM.fireEvent(global, VMM.Timeline.Config.events.data\_ready, \_data\_obj);

},

parseJSON: function(d) {

trace("parseJSON");

if (d.timeline.type == "default") {

trace("DATA SOURCE: JSON STANDARD TIMELINE");

VMM.fireEvent(global, VMM.Timeline.Config.events.data\_ready, d);

} else if (d.timeline.type == "twitter") {

trace("DATA SOURCE: JSON TWEETS");

VMM.Timeline.DataObj.model\_Tweets.buildData(d);

} else {

trace("DATA SOURCE: UNKNOWN JSON");

trace(type.of(d.timeline));

};

},

/\* MODEL OBJECTS

New Types of Data can be formatted for the timeline here

================================================== \*/

model: {

googlespreadsheet: {

getData: function(raw) {

var \_key = VMM.Util.getUrlVars(raw)["key"];

var \_url = "https://spreadsheets.google.com/feeds/list/" + \_key + "/od6/public/values?alt=json";

VMM.getJSON(\_url, VMM.Timeline.DataObj.model.googlespreadsheet.buildData);

},

buildData: function(d) {

VMM.fireEvent(global, VMM.Timeline.Config.events.messege, "Parsing Data");

var \_data\_obj = VMM.Timeline.DataObj.data\_template\_obj;

for(var i = 0; i < d.feed.entry.length; i++) {

var dd = d.feed.entry[i],

dd\_type = "";

if (typeof dd.gsx$type != 'undefined') {

dd\_type = dd.gsx$type.$t;

} else if (typeof dd.gsx$titleslide != 'undefined') {

dd\_type = dd.gsx$titleslide.$t;

}

if (dd\_type.match("start") || dd\_type.match("title") ) {

\_data\_obj.timeline.startDate = dd.gsx$startdate.$t;

\_data\_obj.timeline.headline = dd.gsx$headline.$t;

\_data\_obj.timeline.asset.media = dd.gsx$media.$t;

\_data\_obj.timeline.asset.caption = dd.gsx$mediacaption.$t;

\_data\_obj.timeline.asset.credit = dd.gsx$mediacredit.$t;

\_data\_obj.timeline.text = dd.gsx$text.$t;

\_data\_obj.timeline.type = "google spreadsheet";

} else if (dd\_type.match("era")) {

var \_era = {

"startDate": dd.gsx$startdate.$t,

"endDate": dd.gsx$enddate.$t,

"headline": dd.gsx$headline.$t,

"text": dd.gsx$text.$t,

"tag": ""

};

if (typeof dd.gsx$tag != 'undefined') {

\_era.tag = dd.gsx$tag.$t;

}

\_data\_obj.timeline.era.push(\_era);

} else {

var \_date = {

"type": "google spreadsheet",

"startDate": dd.gsx$startdate.$t,

"endDate": dd.gsx$enddate.$t,

"headline": dd.gsx$headline.$t,

"text": dd.gsx$text.$t,

"asset": {

"media": dd.gsx$media.$t,

"credit": dd.gsx$mediacredit.$t,

"caption": dd.gsx$mediacaption.$t

},

"tag": ""

};

if (typeof dd.gsx$tag != 'undefined') {

\_date.tag = dd.gsx$tag.$t;

}

if (typeof dd.gsx$tag != 'undefined') {

\_date.asset.thumbnail = dd.gsx$mediathumbnail.$t;

}

\_data\_obj.timeline.date.push(\_date);

}

};

VMM.fireEvent(global, VMM.Timeline.Config.events.data\_ready, \_data\_obj);

}

},

storify: {

getData: function(raw) {

//http://storify.com/number10gov/g8-and-nato-chicago-summit

//http://api.storify.com/v1/stories/number10gov/g8-and-nato-chicago-summit

VMM.fireEvent(global, VMM.Timeline.Config.events.messege, "Loading Storify...");

var \_key = raw.split("storify.com\/")[1];

var \_url = "http://api.storify.com/v1/stories/" + \_key + "?per\_page=300&callback=?";

var storify\_timeout = setTimeout(function() {

trace("STORIFY timeout");

VMM.fireEvent(global, VMM.Timeline.Config.events.messege, "Storify is not responding");

}, 6000);

VMM.getJSON(\_url, VMM.Timeline.DataObj.model.storify.buildData)

.error(function(jqXHR, textStatus, errorThrown) {

trace("STORIFY error");

trace("STORIFY ERROR: " + textStatus + " " + jqXHR.responseText);

})

.success(function(d) {

clearTimeout(storify\_timeout);

});

},

buildData: function(d) {

VMM.fireEvent(global, VMM.Timeline.Config.events.messege, "Parsing Data");

var \_data\_obj = VMM.Timeline.DataObj.data\_template\_obj;

\_data\_obj.timeline.startDate = new Date(d.content.date.created);;

\_data\_obj.timeline.headline = d.content.title;

trace(d);

//d.permalink

var tt = "";

var t\_name = d.content.author.username;

var t\_nickname = "";

if (typeof d.content.author.name != 'undefined') {

t\_name = d.content.author.name;

t\_nickname = d.content.author.username + "&nbsp;";

}

if (typeof d.content.description != 'undefined' && d.content.description != null) {

tt += d.content.description;

}

tt += "<div class='storify'>"

//tt += " <a href='" + d.content.permalink + "' target='\_blank' alt='link to original story' title='link to original story'>" + "<span class='created-at'></span>" + " </a>";

tt += "<div class='vcard author'><a class='screen-name url' href='" + d.content.author.permalink + "' target='\_blank'>";

tt += "<span class='avatar'><img src='" + d.content.author.avatar + "' style='max-width: 32px; max-height: 32px;'></span>"

tt += "<span class='fn'>" + t\_name + "</span>";

tt += "<span class='nickname'>" + t\_nickname + "<span class='thumbnail-inline'></span></span>";

tt += "</a>";

//tt += "<span class='nickname'>" + d.content.author.stats.stories + " Stories</span>";

//tt += "<span class='nickname'>" + d.content.author.stats.subscribers + " Subscribers</span>";

tt += "</div>"

tt += "</div>";

\_data\_obj.timeline.text = tt;

\_data\_obj.timeline.asset.media = d.content.thumbnail;

//\_data\_obj.timeline.asset.media = dd.gsx$media.$t;

//\_data\_obj.timeline.asset.caption = dd.gsx$mediacaption.$t;

//\_data\_obj.timeline.asset.credit = dd.gsx$mediacredit.$t;

\_data\_obj.timeline.type = "storify";

for(var i = 0; i < d.content.elements.length; i++) {

var dd = d.content.elements[i];

var is\_text = false;

var d\_date = new Date(dd.posted\_at);

//trace(tempdat);

trace(dd.type);

//trace(dd);

var \_date = {

"type": "storify",

"startDate": dd.posted\_at,

"endDate": dd.posted\_at,

"headline": " ",

"slug": "",

"text": "",

"asset": {

"media": "",

"credit": "",

"caption": ""

}

};

/\* MEDIA

================================================== \*/

if (dd.type == "image") {

if (typeof dd.source.name != 'undefined') {

if (dd.source.name == "flickr") {

\_date.asset.media = "http://flickr.com/photos/" + dd.meta.pathalias + "/" + dd.meta.id + "/";

\_date.asset.credit = "<a href='" + \_date.asset.media + "'>" + dd.attribution.name + "</a>";

\_date.asset.credit += " on <a href='" + dd.source.href + "'>" + dd.source.name + "</a>";

} else if (dd.source.name == "instagram") {

\_date.asset.media = dd.permalink;

\_date.asset.credit = "<a href='" + dd.permalink + "'>" + dd.attribution.name + "</a>";

\_date.asset.credit += " on <a href='" + dd.source.href + "'>" + dd.source.name + "</a>";

} else {

\_date.asset.credit = "<a href='" + dd.permalink + "'>" + dd.attribution.name + "</a>";

if (typeof dd.source.href != 'undefined') {

\_date.asset.credit += " on <a href='" + dd.source.href + "'>" + dd.source.name + "</a>";

}

\_date.asset.media = dd.data.image.src;

}

} else {

\_date.asset.credit = "<a href='" + dd.permalink + "'>" + dd.attribution.name + "</a>";

\_date.asset.media = dd.data.image.src;

}

\_date.slug = dd.attribution.name;

if (typeof dd.data.image.caption != 'undefined') {

if (dd.data.image.caption != 'undefined') {

\_date.asset.caption = dd.data.image.caption;

\_date.slug = dd.data.image.caption;

}

}

} else if (dd.type == "quote") {

if (dd.permalink.match("twitter")) {

\_date.asset.media = dd.permalink;

\_date.slug = VMM.Util.untagify(dd.data.quote.text);

} else if (dd.permalink.match("storify")) {

is\_text = true;

\_date.asset.media = "<blockquote>" + dd.data.quote.text.replace(/<\s\*\/?\s\*b\s\*.\*?>/g,"") + "</blockquote>";

}

} else if (dd.type == "link") {

\_date.headline = dd.data.link.title;

\_date.text = dd.data.link.description;

if (dd.data.link.thumbnail != 'undefined' && dd.data.link.thumbnail != '') {

\_date.asset.media = dd.data.link.thumbnail;

} else {

\_date.asset.media = dd.permalink;

}

//\_date.asset.media = dd.permalink;

\_date.asset.caption = "<a href='" + dd.permalink + "' target='\_blank'>" + dd.data.link.title + "</a>"

\_date.slug = dd.data.link.title;

} else if (dd.type == "text") {

if (dd.permalink.match("storify")) {

is\_text = true;

var d\_name = d.content.author.username;

var d\_nickname = "";

if (typeof dd.attribution.name != 'undefined') {

t\_name = dd.attribution.name;

t\_nickname = dd.attribution.username + "&nbsp;";

}

var asset\_text = "<div class='storify'>"

asset\_text += "<blockquote><p>" + dd.data.text.replace(/<\s\*\/?\s\*b\s\*.\*?>/g,"") + "</p></blockquote>";

//asset\_text += " <a href='" + dd.attribution.href + "' target='\_blank' alt='link to author' title='link to author'>" + "<span class='created-at'></span>" + " </a>";

asset\_text += "<div class='vcard author'><a class='screen-name url' href='" + dd.attribution.href + "' target='\_blank'>";

asset\_text += "<span class='avatar'><img src='" + dd.attribution.thumbnail + "' style='max-width: 32px; max-height: 32px;'></span>"

asset\_text += "<span class='fn'>" + t\_name + "</span>";

asset\_text += "<span class='nickname'>" + t\_nickname + "<span class='thumbnail-inline'></span></span>";

asset\_text += "</a></div></div>";

\_date.text = asset\_text;

// Try and put it before the element where it is expected on storify

if ( (i+1) >= d.content.elements.length ) {

\_date.startDate = d.content.elements[i-1].posted\_at;

} else {

if (d.content.elements[i+1].type == "text" && d.content.elements[i+1].permalink.match("storify")) {

if ( (i+2) >= d.content.elements.length ) {

\_date.startDate = d.content.elements[i-1].posted\_at;

} else {

if (d.content.elements[i+2].type == "text" && d.content.elements[i+2].permalink.match("storify")) {

if ( (i+3) >= d.content.elements.length ) {

\_date.startDate = d.content.elements[i-1].posted\_at;

} else {

if (d.content.elements[i+3].type == "text" && d.content.elements[i+3].permalink.match("storify")) {

\_date.startDate = d.content.elements[i-1].posted\_at;

} else {

trace("LEVEL 3");

\_date.startDate = d.content.elements[i+3].posted\_at;

}

}

} else {

trace("LEVEL 2");

\_date.startDate = d.content.elements[i+2].posted\_at;

}

}

} else {

trace("LEVEL 1");

\_date.startDate = d.content.elements[i+1].posted\_at;

}

}

\_date.endDate = \_date.startDate

}

} else if (dd.type == "video") {

\_date.headline = dd.data.video.title;

\_date.asset.caption = dd.data.video.description;

\_date.asset.caption = dd.source.username;

\_date.asset.media = dd.data.video.src;

} else {

trace("NO MATCH ");

trace(dd);

}

if (is\_text) {

\_date.slug = VMM.Util.untagify(dd.data.text);

}

\_data\_obj.timeline.date.push(\_date);

};

VMM.fireEvent(global, VMM.Timeline.Config.events.data\_ready, \_data\_obj);

}

},

tweets: {

type: "twitter",

buildData: function(raw\_data) {

VMM.bindEvent(global, VMM.Timeline.DataObj.model.tweets.onTwitterDataReady, "TWEETSLOADED");

VMM.ExternalAPI.twitter.getTweets(raw\_data.timeline.tweets);

},

getData: function(raw\_data) {

VMM.bindEvent(global, VMM.Timeline.DataObj.model.tweets.onTwitterDataReady, "TWEETSLOADED");

VMM.ExternalAPI.twitter.getTweetSearch(raw\_data);

},

onTwitterDataReady: function(e, d) {

var \_data\_obj = VMM.Timeline.DataObj.data\_template\_obj;

for(var i = 0; i < d.tweetdata.length; i++) {

var \_date = {

"type":"tweets",

"startDate":"",

"headline":"",

"text":"",

"asset":

{

"media":"",

"credit":"",

"caption":""

},

"tags":"Optional"

};

// pass in the 'created\_at' string returned from twitter //

// stamp arrives formatted as Tue Apr 07 22:52:51 +0000 2009 //

//var twit\_date = VMM.ExternalAPI.twitter.parseTwitterDate(d.tweetdata[i].raw.created\_at);

//trace(twit\_date);

\_date.startDate = d.tweetdata[i].raw.created\_at;

if (type.of(d.tweetdata[i].raw.from\_user\_name)) {

\_date.headline = d.tweetdata[i].raw.from\_user\_name + " (<a href='https://twitter.com/" + d.tweetdata[i].raw.from\_user + "'>" + "@" + d.tweetdata[i].raw.from\_user + "</a>)" ;

} else {

\_date.headline = d.tweetdata[i].raw.user.name + " (<a href='https://twitter.com/" + d.tweetdata[i].raw.user.screen\_name + "'>" + "@" + d.tweetdata[i].raw.user.screen\_name + "</a>)" ;

}

\_date.asset.media = d.tweetdata[i].content;

\_data\_obj.timeline.date.push(\_date);

};

VMM.fireEvent(global, VMM.Timeline.Config.events.data\_ready, \_data\_obj);

}

}

},

/\* TEMPLATE OBJECTS

================================================== \*/

data\_template\_obj: { "timeline": { "headline":"", "description":"", "asset": { "media":"", "credit":"", "caption":"" }, "date": [], "era":[] } },

date\_obj: {"startDate":"2012,2,2,11,30", "headline":"", "text":"", "asset": {"media":"http://youtu.be/vjVfu8-Wp6s", "credit":"", "caption":"" }, "tags":"Optional"}

};

}