All data and functions exist in a global object accessible as *ReactiveSpaces* or simple *RS*.

# Settings

**RS.BASEURL**

Stores the base host/url for connecting to the desktop client. In most cases, this should not be changed. If you wanted to try launching the desktop app on another PC and then use its Kinect, it should theoretically work by changing this address.

*Default value* : "ws://localhost"

**RS.LOCALPORT**

Stores the port on which to connect to the desktop app. Only change this value if you needed to change the connection port in the desktop app.

*Default value* : 8080

**RS.MESSAGE\_DELAY**

Stores the number of milliseconds delay required between each sent message. Shortening this value could cause instability and lag in the system and server. If you’re game requires messages more often than this, you should look at using a true multiplayer system.

*Default value* : 100

RS.SUPORTED\_PLAYERS = 2;

Represents the maximum number of local players that will ever be tracked. Changing this value has no effect.

*Default value* : 2

**RS.MOVEMENT\_BLENDING**

Boolean value denoting weather or not to blend skeleton frame data and create smoother joint movement. Turning this off can improve performance, but blended joint positions will no longer update.

*Default value* : true

# Members

**RS.socketSupported**

Stores weather or not the browser supports web sockets. If false, Reactive Spaces will not be able to connect to the desktop app or server.

**RS.socket**

Stores the current socket connection to the desktop app (if any).

**RS.connected**

Boolean value representing the current connection status of Reactive Spaces.

**RS.lastMessage**

UNIX timestamp denoting the last time a message was sent from Reactive Spaces to the desktop app.

**RS.station**

Stores the *RS.StationProfile* object representing the local station information. If no connection to the desktop app has been made, this value can also be null.

**RS.remoteStations**

Stores a list of *RS.StationProfile* objects representing the remote stations that are in the same game session.

**RS.players**

Stores a list of *RS.Skeleton* objects representing the current local players.

**RS.remotePlayers**

Stores a list of *RS.Skeleton* objects representing the current remote players.

**RS.lastBlendUpdate**

UNIX timestamp denoting the last time the blended skeleton data was updated

**RS.listeners**

Stores all event listeners added to Reactive Spaces

**RS.messenger**

Provides basic error logging capabilities for Reactive Spaces.

Also defines a Global *Message* object which has an enumerator for message types defined as

*Message.type = {“MESSAGE” : 0, “WARNING” : 1, “ERROR” : 2}*

**RS.appInfo**

Stores information about the connected web app. Members are: name, version, maxPeers.

Default Value:*{*

*name: "Default App Name",*

*version: 0,*

*maxPeers: 4*

*}*

# Methods

**RS.Connect** ( appName, appVersion, features, port )  
*appName* : name of web app that is connecting  
*appVersion* : version of app that is connecting (int or float). Only app instances with the same version will be matched on the server   
*features* : list of features that are required by the app. Use enumerator *RS.Features.*  
*port* (optional):connection port to use. If omitted, RS.LOCALPORT will be used

Attempts to connect to the Reactive Spaces Desktop App. Returns true or false depending on whether connection was successful or not.

**RS.Disconnect** ()

Disconnects from the desktop app and server.

**RS.BlendUpdate** ()

Updates the blended skeleton positions. Called automatically when *RS.MOVEMENT\_BLENDING* is true on connection. Can be called manually on your apps update loop if desired.

**RS.ActivateMessenger** ()

Reactive Spaces provides an optional messenger system to display messages, warnings and errors easily.

Include the messenger in you project with

*<script src=” http://reactivespacesapi.com/lib/Messenger.js”></script>*

And then call this function to upgrade *RS.messenger* to this advanced messenger. Post your own messages with *RS.messenger.display( Messenge.type type, string message, string solution).*

**RS.addEventListener** (event, callback)  
*event* : string denoting the envent to listen to. Use enumerator RS.Events or see RS.Events in the documentation for a full list  
*callback* : function pointer for function to be called when event fires

**RS.removeEventListener** (event, callback)

This function currently has no effect

**RS.Send** ( object )  
*object*: javascript object to send

Sends the given object to all stations in the current app session.

# Types

# Enumerators