

New DataChannel implementation details – version 0.8.868 and before

Documentation deprecated

This documentation was updated for versions 0.8.869 and above. Please visit the [new documentation](#).

Abstract

Due to limitations in the plugin interface, we had to delegate part of the data sending implementation to javascript. This document explains how this was done, and what the repercussions of this solution are.

Sending a string

If you use the plugin to send a string with this call:

```
var data = "My string data";
mySendingChannel.send(data);
```

Then the plugin is directly sending the string "data" to the other side of the data channel.

Sending a non string data

If you send one of the following types:

- Int8Array
- Uint8Array
- Int16Array
- Uint16Array
- Int32Array
- Uint32Array

```
var data = new Int8Array(10);
mySendingChannel.send(data);
```

then the plugin will call the following javascript function to parse the ArrayBuffer into a string that can be sent to through the plugin interface

```
_Tem_ab2str = function(buf) {
    var str = String.fromCharCode.apply(null, new Uint8Array(buf.buffer));
    return new Uint8Array(buf.buffer);
};
```

When the remote data channel receives the string, it will change it back into a Uint8Array using the following function:

```
_Tem_str2ab = function(str) {
    var buf = new ArrayBuffer(str.length);
    var bufView = new Uint8Array(buf);
    for (var i=0, strLen=str.length; i<strLen; i++) {
        bufView[i] = str[i];
    };
};
```

When you receive the data on the other side of the data channel, you can then use:

```
function onReceiveMessageCallback(event) { // was set as the receiving
    datachannel onmessage callback
        var data = new Int8Array(event.data);
        ...
    }
```

Temasys considered it the best available solution and does not plan on releasing a different implementation in the close future.

Sending other types of data

Sending other types of data will lead to unexpected behaviour. Your data might be sent successfully, some of it might be lost or corrupted, or the plugin could even crash.

Implications

1. The implementation of string sending is that same as it was before.
2. You should not override the functions `_Tem_ab2str` nor `_Tem_str2ab`.
3. Overriding these functions might lead to unexpected behaviours

Examples of how to use the data channel

We adapted the `dataChannel` sample provided by Google to work with the plugin, and implemented a second test for sending non-string elements.

Please refer to these examples if you try to implement a data channel in your own solution.

DataChannel and strings: <https://github.com/Temasys/Google-WebRTC-Samples/tree/master/samples/web/content/datachannel>

DataChannel and ArrayBuffers: <https://github.com/Temasys/Google-WebRTC-Samples/tree/dev/samples/web/content/datachannel-arraybuffer>