Ecma/TC39/2012/004

Binary Data Updates

Binary Data

- **Goal**: Maximal consistency and interop with Typed Arrays, address concern about access to underlying buffer.

ArrayTypes not fixed length

```
// Using ArrayType in Typed Array scenarios
var Int8Array = new ArrayType(Int8);
// Same as Typed Arrays
var arr1 = new Int8Array(100);
var arr2 = new Int8Array(256);


// Using ArrayType in structs
var MyStruct = new StructType({
    // Change from Binary Data, pass length to ArrayType
    // Somewhat closer to C structs
    // Note: [[Call]] and [[Construct]] have different behaviour
    extra: Int8Array(100)
})
```

ArrayType instance constructor overloads

```
[ Constructor(unsigned long length),     // allocate
  Constructor(type[] array),             // initialize
  Constructor(TypedArray array),         // copy
  Constructor(ArrayBuffer buffer,        // view over buffer
              optional unsigned long byteOffset,
              optional unsigned long length)
]
interface ArrayType {

}

// ES-spec writeup:
```

Access to underlying ArrayBuffer

```
var MyUint8Array = new ArrayType(uint8);
var uints = new MyUint8Array(100);
uints.buffer //Does this behave the same as type arrays?

// Concern about exposing endianness, struct layout
// Ship has partly sailed with Type Arrays + file i/o requires
// Proposal: Only the arraybuffer overload exposes buffer
var uints1 = new MyUint8Array(100);
uints1.buffer === null;
var buf = new ArrayBuffer(100);
var uints2 = new MyUint8Array(buf);
uints2.buffer === buf
```

String support in Binary Data

- String encode/decode over typed arrays in Web WGs.
- **Proposal**: No string support directly in binary data. Use host string libraries (web platform, server host, etc.)