

**Minutes of the:**

**Ecma TC39, ES3.1WG**

**Phone conference**

**held on:**

**08 April 2008**

## 1 Roll call and logistics

### 1.1 Participants

Mike Cowlshaw (IBM), Doug Crockford (Yahoo!), Jeff Dyer (Adobe), Lars Hansen (Adobe), Collin Jackson (Stanford), Pratap Lakshman (Microsoft), Mark Miller (Google), Adam Peller (IBM), Wayne Vicknair (IBM), Allen Wirfs-Brock (Microsoft) and Kris Zyp (The Dojo Foundation).

## 2 Agenda

Decimal support in ES3.1 and ES4

## 3 Minutes

Mike provided a background and summary on decimal, and where we are now in terms of implementations - decNumber reference implementation (in ANSI C), support in GCC, BigDecimal in Java, support in C# and the CLI - it is supported in most common and important languages - it ought to be supported in ECMAScript too - issues with implementation divergence and performance issues - but that is being resolved now through the IEEE754 revision (754R).

But Java's support in library only - would that be a workable solution for ECMAScript ? - it would be workable but not the best solution per Mike.

The challenge for ES3.1 is how to introduce it in an evolutionary style - can we let implementations choose, at their discretion, the representation for the number type ? - unlikely to find adoption in the browser embeddings - which means nobody will implement it anyway.

Need to reduce the 'distance' between ES3.1 and what the browsers implement today - adding decimal support in ES3.1 would increase that distance - ES3.1 has a goal of harmonizing existing implementations that have diverged.

Is it just a spec-writing issue ? - is it only going to add complexity for the implementer ? - no, not quite - ES3.1 needs to reduce the semantic complexity of the spec - need to reduce the burden on the user - he should not have to worry about mixing and matching doubles and decimals and the consequent coercions, etc.

ES4 intends to introduce decimal as a first class type with literals to denote the type, operator support, and means of controlling rounding and precision - ES3.1 needs to be careful about maintaining compatibility.

Given that ES3.1's scope is different than ES4 how about decimal support being there only in ES4 and not in ES3.1 - no, IBM would like to see decimal support be available in implementations as soon as possible - given the ES3.1 implementations are likely to come into existence sooner than those for ES4, decimal support ought to be there in both ES3.1 and ES4.

Is the current representation of decimal in the CLR an acceptable implementation of the IEEE standard ? - there are two versions of the standard out there - the CLR implements a smaller

subset of the standard and, strictly speaking, cannot be considered a compliant implementation.

How about saying that ES3.1 support for decimal is not compatible with the ES4 support for the native decimal data type ? - sounds reasonable - not really, that could cause the testing burden to increase.

How about both ES3.1 and ES4 provide support through a library (and not at the language level) ? - that is not acceptable to ES4 - ES4 would prefer to support it as a first class data type - there is a certain aesthetic to ES4, and it would look weird if it supported it both as a library and at the language level.

Lets think about these issues some more, and reconvene on this topic a few weeks from now - for the time being we can continue discussing this topic over email with the present set of people - once an "ES3.1 discuss" mailing list is created we can move the discussion there.

Meeting adjourned.