

**Minutes for the:** *8<sup>th</sup> meeting of Ecma TC39*  
**in:** *Mountain View, CA, USA*  
**on:** *28-29 January 2009*

**Chairman:** Mr. John Neumann (Microsoft/Ecma International)  
**Vice-Chairman:** Vacancy  
**Secretary:** Mr. John Neumann (Adobe/Microsoft/Ecma International)  
**Attending:** Mr. Ihab Awad (Google), Mr. Jeff Dyer (Adobe), Mr. Brendan Eich (Mozilla), Mr. Cormac Flanagan (UCSC), Mr. Dave Herman (Northeastern University), Mr. Waldemar Horwat (Google), Mr. Blake Kaplan (Mozilla), Mr. Kris Kowal (Fast Soft (invited guest), Mr. Pratap Lakshman (Microsoft), Mr. Waldemar Horwat (Google), Mr. Mark S. Miller (Google), Mr. Sam Ruby (IBM), Mr. Rob Sayre (Mozilla), Mr. Kwang Yui Seo (Company 100) and Mr. Allen Wirfs-Brock (Microsoft).

## 1 Opening, welcome and roll call

Meeting opened at approximately 10:15 am by the chairman

### 1.1 Host facilities

Waldemar gave a briefing on facilities for the day and plans for lunch and dinner.

### 1.1 Dinner arrangements

A good time was had by all and appreciation was expressed to Ecma for hosting the dinner.

### 1.1 Lunch

Lunch provided daily by Google.

## 2 Adoption of the agenda ([09/003](#))

Adopted as presented

## 3 Approval of the minutes of November 19<sup>th</sup> and 20<sup>th</sup>, 2008 ([08/105](#))

Approved as written

## 4 ES3.1 draft proposal discussion

Need to remove ToObject from base when making references:

```
function foo() {return this} (in strict mode)
```

```
foo.call(3) -> returns a primitive (i.e. non-wrapped) 3
```

```
String.prototype.foo = foo
```

```
"bar".foo() -> needs to return a primitive too
```

Question: what if the function does:

```
function foo() {this.x = 5}
```

Answer: in non-strict mode, works as before. In strict mode, throws an exception due to an attempt to set a property on a primitive.

Use a `:::` instead of `:` grammar to describe use directive syntax in chapter 14. There are other examples of such usage in chapter 15 that can be used as a guideline. Also should describe white space using the lexer grammar, not the syntactic grammar.

Outstanding Tickets review and all issues resolved

1. `[[class]] = 'string', 'date', 'regexp', 'array', 'function'`

Need `[[isArray]]`? Is `[[Class]]` trustworthy?

Should we enforce constraints on host objects of these classes?

Strawman: Host objects should only be able to do what native objects can do -- doesn't work for host functions that can access data outside the ECMAScript environment.

Many of the invariants are enforced by native constructors' behavior, not by constraints on native objects.

Proposal:

+ Change instances of "is an Array object" to "`[[Class]]` is "Array"", etc.

+ Write specifications on invariants that host objects must satisfy in order to use one of the above `[[Class]]` values. Who will do it?

Dealing with multiple global objects: Informative annex to 3.1? A Technical Report would be more appropriate.

2. Bind curry over `[[construct]]`? How? Should 'bind' result in frozen funct? Agreement to accept option 2 from Allen.

- Agreed to use option 2 from Allen's list: bind will curry over `[[Call]]` and `[[Construct]]`. The bound closure will not have a prototype property -- `[[Construct]]` will use the prototype of the original function just as it normally would.

- Agreed that bind does not result in a frozen function.

3. When is eval direct? Strict?

- Agreed to keep the current two conditions of a direct call named "eval" and the function actually evaluating to the built-in eval function.

- Agreed to disallow the use of "eval" as the name of a local variable, function parameter, etc. in strict mode.

- Agreed that eval is strict if and only if the eval'd string starts with a use directive.

- Agreed that indirect eval calls will use their own little private scope for newly introduced bindings in strict mode (as determined by whether the eval'd string starts with a use directive), even if they're called at the global level.

4. Should 'apply' be specified like an array generic

- Agreed that it should be generic. This will also make it explicit what happens if there are holes in an array.

5. Should numerically named props of strings be enumerable?

- Yes.

6. Parse time error reporting ('strict')

- "detection does not require program execution" is too vague. Agreed to retain the "must" as long as individual cases are listed instead of "detection does not require program execution".

7. Regexp look ahead

- Rejected the ticket. Rejecting quantifiers syntactically would be a breaking change and make grammar syntax more complicated. There is a use for quantifiers after lookaheads -- executing a quantifier zero times is different from executing it once, and the differences can be observed via capturing parentheses.

#### 8. Name property of getter/setter functions in literals

- Property values for function <name> are "get <name>", "set <name>", "bind <name>". They do not show up on the scope chain.

#### 9. Multiline comment

Multiline comments:

```
break/*  
*/foo;
```

See <http://bugs.ecmascript.org/ticket/414>

- Agreed to retain the specified ECMAScript behavior: a multiline comment counts as if it were a line terminator.

11.1.4: Replace [[Put]] of length with DefineOwnProperty. Generic array algorithms will continue to be done using [[Get]] and [[Put]] (this is important for folks who define length properties as getter-setter pairs).

JSON throws a SyntaxError when it detects a problem with the input. This is the first instance of throwing a SyntaxError based on what's typically user input data. Suggestions were made for using a FormatError or JSONError. Waldemar reluctantly relented to reach consensus to keep SyntaxError, mainly due to compatibility with existing JSON libraries. This means that users who catch SyntaxErrors hoping to catch JSON errors will catch things that are not user data errors as well if, for example, the ECMAScript program constructs faulty regular expressions or eval's code with syntax errors.

What causes the arguments object to get un-joined with local argument variables?

- Allen's current approach (changing enumerability or configurability doesn't matter, changing value breaks joining) seems like the right approach.
- Agreed that in strict mode there is no joining of arguments objects to local variables.
- Agreed that arguments objects are \*not\* frozen in strict mode. There are important use cases for mutating them, such as stripping off initial arguments using shift and passing the rest to another function.

Further discussion on decimal. Implementation and testing will be aimed at inclusion in Harmony, and will be done in sufficient time (over the next couple of months) for inclusion in that version of ES.

## 4.1 Testing

Started, but not sufficient progress to keep the original schedule of being finished in June GA.

## 4.2 Schedule and planned completion

John and Allen want a flag date to issue s press release about a candidate spec. Change control may be locked down after that ?

Waldemar: Ecma should not be the editor. Changes need to be made by a technically savvy editor.

Waldemar's concern: document has been evolving too quickly, and each time he reviews the document he needs to re-read the whole thing which takes a long time. Proposed two levels of change bars after a settle-down date, with changes from ES3 -> settle-down date tracked with a different color from changes settle-down date -> head revision.

Agreed to a new schedule as shown below.

Jan to March: Final document reviews and preparation of Candidate Specification

March: Document effectively frozen

March to July: Testing and review and fixing problems

July: Testing complete, Document review complete

July to September: Document revised and polished for publications

September: Final approval and submission to CC/GA

## 5 ES4 draft proposal discussion

### 5.1 Presentation by Ihab Awad (with Kris Kowal [Fastsoft]) on Proposed Harmony module

Presentation was given and is available on TC39 web site. There was substantial discussion, with no decisions taken.

### 5.2 ES 4 Discussion of base line content and extensions

### 5.3 ES 4 Schedule

## 6 Progress reports on the "Secure ECMAScript" Plan

Project started out to define a competition environment for defining the best SES capability and solution. It is currently thought that a merger of several proposals (from Doug and Mark) with minor change to 3.1 might get a solution quicker than the previous plan. In addition, it is felt that changes to technical elements owned by W3C necessitates a close liaison with that organization. Requirements will be defined and vetted within the broader TC39 community before sending to W3C. Also we have been invited to hold joint meetings with W3C in November, and when they meet in Santa Clara, California. To that end we have tentatively scheduled our November meeting at the same time and will investigate meeting in the same location (Santa Clara Marriott, and the chair offered for Ecma to pay for the cost of the meeting room)

## 7 Any other business

Brief demonstration of ES 3.1 from Microsoft

## 8 Date and place of the next meeting(s)

- March (Mountain View, California) [24-25-26] @ 10 AM [Yahoo]
- May (Mountain View, California) [27-28-29] @ 10 AM [Mozilla?]
- July (Redmond, Washington) [28-29-30] @ 10 AM [Microsoft]
- September (Bay Area, California) [15-16-17] @ 10 AM (Google)
- November (Santa Clara, California) [3-4-5] @ 10 AM (Ecma – co located with W3C)

## 9 Closure

Meeting was closed at approximately 4 PM on Thursday, January 29, 2009