Ecma/TC39/2008/051

Minutes of the: ***Ecma TC39, ES3.1WG***

held in: ***Phone conference***

on: ***24 June 2008***

1 Roll call and logistics

1.1 Participants

Pratap Lakshman (Microsoft), Mark Miller (Google), Adam Peller (IBM), Sam Ruby (IBM) and Allen Wirfs-Brock (Microsoft)

2 Agenda

On posting the latest draft to the wiki

Getters/Setters

Decimal

Setting up a review based on Lars' feedback on the 11 June draft

3 Minutes

Would like to add a couple more items to agenda that we can get to if we have the time (1) inconsistence language like "as if by the expression …" pervasive in ES3 (e.g. section 11.1.4 Array Initializer: "create a new array as if by the expression new Array()"; needs to be fixed in ES3.1 (2) ES3/ES4 based review.

On posting the latest draft to the wiki

The latest draft has been uploaded to the wiki. This the draft as of 24 June 2008 - it has all the edits related to the statics on Object, the introduction of the "Extensible" property, the revised notation for JSON, and placeholders for Decimal - by next week we should try to have a technically complete draft - the draft may still have some place holders but it should still be enough to get circulated within TC39 as an artifact that can be reviewed - by the time we meet in Oslo, each place holder must have a supplementary doc that can be discussed F2F.

'const' needs to be introduced into the grammar - what does it mean to be a reserved word ? - you cannot use it as a variable name - note that we have introduced the ability to use is as the name of a property though – const has letrec like scoping – not subject to hoisting.

reformed scoping needs to be introduced - but for doing that we may want to introduce the notion of a 'block activation' as an expository device

'abc'[i] is using the ES3 like algorithm convention – can we use the new convention that eliminates the need for writing gotos – pratapL to investigate – also, for all new functionality we need to worry about argument conditioning – for e.g. if an algorithm expects to be handed an object, make sure we call toObject on the argument that is passed in.

Getters/Setters

Surface syntax spec from Kris looked Ok - need to ensure it is using the Meta APIs - Allen to check surface syntax integration.

Decimal

Decimal changes are isolated and can be done without impacting content in the rest of the doc - Sam can use the latest draft from the wiki as a starting point.

Strict mode

2 controversies coupled to each other – proposed ES4 wants to keep ‘with’ in their strict mode; ES3.1 wants to remove ‘with’ in its strict mode – need to be clear about the purpose of strict mode – the other is “what do we mean by subset ?” – there was one formal definition from Lars, and Doug had called out a less formal notion – we may end up using Doug’s less formal notion (especially if we cannot resolve the ‘with’ issue).

Binary strict mode is naively limiting – each version may want to allow/limit specific features – strict mode allows users to say ‘I am subscribing to this particular subset, and I am aware of all its limitations’ – but if the subset is a moving target how do they do that ? - sure, we can introduce a more elaborate mechanism – it would make it easier for us to not have to resolve arguments – but it would open up a larger combinatorial space of possibilities – the SunOS vs. MacOS problem; the former was highly customizable, however you invariably got it wrong; the latter was not customizable, but what you got was good enough to get the job done – more knobs need not mean better.

An elaborate mechanism enables chaos at the composability boundary – but how does it matter if you are opaquely including a module? – Caja will use ES3.1 strict mode for all uncajoled code – OTOH an elaborate mechanism can enforce a subsetting profile – languages can now more directly enforce Caja like semantics.

‘use strict 3.1’ could be a potential syntax to turn on 3.1 level strict mode – with the possibility that the 3.1 may be replaced with a list – instead of tying it to the language version number, we can also just say something like ‘use strict a’ or ‘use strict 1’ – can also use Perl-style ‘use strict no with’ (where we mention the specific restriction we want to enable) – that seems a good idea too – in any case, named restriction sets for ES3.1 and proposed ES4 will be useful to discuss this proposal further.

Meeting adjourned.