Ecma/TC39-TG1/2006/030

Minutes of the: Ecma TC39-TG1

held in: Phone conference

on: 12th July 2006

Attendees

- Jeff Dyer, Adobe Systems
- Douglas Crockford, Yahoo!
- Pratap Lakshman, Microsoft
- Dave Herman, Northeastern University
- Brendan Eich, Mozilla Foundation
- Graydon Hoare, Mozilla Foundation
- Francis Cheng, Adobe Systems

Agenda

- Dave's type conversion proposal
- Nullability
- Continuations
- Date literal syntax

Discussion

Jeff's notes

- Type conversions
  - type conversions don't have the same propoerties as sub typing
  - this causes some confusion; we need to be clear that they are different
  - how to implement type conversions, automatic or explicit?
- Nullability needs to be discussed; homework: review es4-discuss thread and wiki to prep for f2f
- Continuations
  - There has been discussion on the list to provide more general continuations than are currently proposed
  - The group sentiment seems to be that we should not over power the language and to limit yields to span one call frame
- Brendan: we should removed the CF characters
- Date literals
  - Pratap wonders whether we should implement the ISO date syntax for literals
  - Brendan sugguests implementing Lars' current proposal
  - Pratap to post link to C# syntax on wiki

Francis' notes

date literal syntax. Lars is the author of that proposal. May be better to wait until next week to discuss this.

type conversion proposal

How typed and untyped code interact with each other. This is not about subtyping, and this has caused some confusion around this topic. One difference between conversion and subtyping is that conversion is not transitive, as subtyping is.

One way to implement type conversion is to automatically convert known type conversions.

Another way is to make it opt-in via an operator or constructor. User would have to explicitly call for conversion.

Yet another way it to refuse to allow certain types of conversions like those involving function types or anything that is mutable. Would require a wrapper in these cases. Problem is that a function is something with an infinite number of inputs and outputs, so you can't create a table. You have to use a wrapper.

Dave suggests that we don't leave it entirely up to the user. We should provide the wrappers. It would make it hard to use or migrate legacy code. Ideally you'd allow users to script without types, then incrementally add types. If we don't provide wrappers for them, they'll never move to types.

Dave has concerns about how this will interact with prototypes. Maybe it can just treat the prototype link as another field. Jeff thinks this is not a good way to go. Jeff has E4X example. You have a node and add it to another node as a child and you get a copy of it. Spec is fixed, but implementations aren't on this topic.

Dave would like to get Lars to comment on this. Jeff suggests discussing this at the face to face.

Jeff's example:

```
var o = { i : 10 }
type T = { i : int }
var x : T = o     // error or not? explicit conversion required?
```

Jeff: what about duck typing scenario?

Brendan: the real question is the how hard is it to implement solution 2.

Jeff: We should all do our homework and talk about this in depth at the face to face. Another issue that we should nail down is nullability.

Brendan: Charged up to argue against nullability by default.

Dave agrees: It's much more consistent and fix more errors statically.

Jeff: My final public thought is that we should try to not break AS3.

Brendan: Tucker's thought is that there's a latent null dereferenced bug. In AS3 you can declare a var and count on it to be null until it is assigned to. There could be some incompatibility there.

Brendan: Nicolas Cannase also brought up opening up the black box on continuations.

Dave: Folks I've talked to wonder what you really get with full continuations.

Brendan: Someone also brought up adding Lua-like yield functionality.

Dave: It's easier to type in pythonic yields than in full yields.

Brendan: We need to stop stripping control chars. Because complain about the verbosity of lambda in JS. Brendan got a request to add more concise way to write it. Something like "func" or even a single character? Once we have macros, this is generally solved.

Date literal:

Pratap: We could go full ISO or some light version. Just want to get how everyone feels about it.

No one has tried to implement Lars's proposal. Brendan thinks we need to in order to say more.

Douglas: There is already so much overloading of tokens. In this case we'll be overloading slash.

Was this note about overloading slash accurate, or was Doug talking about general lexical complexity? There is no slash in Lars's ISO subset proposal.

— Brendan Eich 2006/07/19 10:16

Brendan: You could be right, we'll need to try implementing it.

Douglas: Could also envision code that accidentally creates a Date literal.