

**Minutes of the:
held in:
on:**

**Ecma TC39, ES3.1WG
Phone conference
08 July 2008**

1 Roll call and logistics

1.1 Participants

Doug Crockford (Yahoo!), Pratap Lakshman (Microsoft), Mark Miller (Google), Sam Ruby (IBM), and Allen Wirfs-Brock (Microsoft)

2 Agenda

ES3.1 global object reference changes

Decimal

3 Minutes

Want to virtualize the behaviour of a readonly property - how do you emulate readonly-ness - by removing the setter - virtual property with a getter but no setter is readonly - no way for a setter to emulate a readonly property - also, cannot detect the strictness of the caller - if a "readonly" attribute is set you never call the setter - then apply strictness on the caller side - setter can be a function or undefined, and then a Writable bit that can be ON or OFF - confusing - if a setter has never been specified, its value must be undefined - ES4 has no way to distinguish a setter that was never specified.

Why do we need `getProperties` and `getOwnPropertyNames`? - can be implemented in JavaScript itself - data properties and accessor properties manifest themselves in object literals in syntax - data properties on an object can be defined implicitly through assignment, and on objects a set of methods that allow you to define either kind of properties with full control - no need for `getProperties` - `getOwnPropertyNames` is a nice complement to `defineOwnProperty` - use case: Caja one time startup gets the global object enumerating everything that is Caja-reachable and locks it down - can still do it one property at a time - no ! cannot know what the properties are before hand - cannot use 'keys' for this either (that gives only enumerable own properties).

Regarding this getting promoted to the global object in the case it is null or undefined, remove all promotions on the setting side, and change 11.1.1 to read "if this is evaluated within an execution context that is subset restricted to the cautious subset and the this value is null or undefined, then the this expression throws a ReferenceError exception". In the non cautious subset, we want the same behaviour as ES3. Also, will need to change the internal "CanPut" method.

Decimal

Lets discuss this at the next meeting.

Meeting adjourned.