4/16Working group meeting notes

Guy's comments on changes he made recently:

* Everyone should re-read the entire native objects section. There have been substantial edits in it. Not all changes are listed in Appendix D, since the changes were so comprehensive.

* The description of Date was changed so that it uses some building blocks.

Date comments/changes:

* 15.9.4.26 For setSconds(...) should undefined and not specified do the same thing? This can be fixed by just saying "if ms is unspecified" rather than "if ms is undefined or not specified"

* 15.9.4.26 The explicit treatment of NaN may be wrong. We either need to fix this explicitly or remove the special treatment (and ensure that the right thing indeed does happen).

* More generally, we need to specify (implicitly or explicitly) what occurs when arguments are omitted.

* 15.9.2.1. The Date constructor should also take a milliseconds parameter.

* The Date constructor will have the year 2000 problem, and we need to state this -- if you pass 0-99 for the year, we add 1900 to it and use that as the year.

* 15.9.14 Step 2 refers to Result(2). This should be Result(1) instead.

* 15.9.2.5 We should use the internal algorithm rather than saying that the implementation actually calls Date.Parse. There are subtle cses in which these are not the same.

* Date.parse needs to handle what is produced by Date.toString and Date.toGMTString. Other than that we are likely to say that the rest of the semantics are implementation-dependent. We can try to nail this down in a later version.

* There is an accidental double-negative in 15.9.4.

Object model comments/changes

* Unless otherwise specified, if a parameter is left out, then it is treated as undefined. Then the normal machinery crunches, and usualy generates a NaN.

* new Function for something that does not parse should generate a runtime error. (This is a change from what we decided earlier.) This will cause a chane to 15.3.2.1 step 16.

* There are things that are r/w today, e.g., String.prototype, that are referenced in other places. Do the changes made by script code affect these other places or not? E.g., in 15.2.2.1 step 7 we refer to the "String prototype object". Does this mean the original one or the new one? Resolved: we will make the prototype properties of built-in ecmascript objects ReadOnly, DontDelete.

* 15.2.2.1 step 7 shouldn't say "exactly like the expression . .

." since this isn't true for cases in whcih String has been altered.

* 15.2.2.1 Better to say new Object(...) is undefined if an external object is passed in. A related change will be needed in 15.2.4.3.

* To do items

* parseFloat(string)

* parseInt(string)

* escape(string). What we have today does not deal with unicode characters. If the high byte it non-zero, we will do %uXXXX unless we find that the RFC in question has already solved this problem.

* unescape(string)

* 15.3.4 var x = Function() should be synonymous with var x = new Function() and var x = new Function("")

* We decided to remove the ErrorOnWrite attribute. It appears twice in the spec today, once in a definition and once in an algorithm. No properties have this attribute so we do not need it..

* The same "missing element" issue brought up in reverse also applies to sort. We agreed that for sort, non-existentant and undefined elements end up at the end. The length property won't change for either sort or reverse. Agreed not to specify whether an item exists with value undefined or doesn't exist.

Comments/changes on the intro and other sections

* Lots of small comments. I didn't take notes during this time.

Next meeting

* We made a fairly large number of changes in this meeting, and our deadline is rapidly approaching.

* Guy is going to find otu what our tiem constraints are, and then we will decide when to meet, and whether we do so by phone or in person.

--Scott