

TC39 TG1 E4 Meeting Notes 2003-06-19

Date	June 19, 2003 10:00 AM – 5:00 PM
Location	Conference call 866-500-6738 access code 795181
Convener	Rok Yu (Microsoft)
Editor	Waldemar Horwat (Netscape)
Participants	Jeff Dyer (Macromedia) John Schneider (BEA/AgileDelta) Rok Yu (Microsoft) Waldemar Horwat (Netscape) Markus Scherer (IBM)

Questions on Algorithms

Jeff states that on reviewing the algorithms for ObjectLiterals, he believes there is a bug in constructObject. He asks if constructObject has a bug where (40) it should create a new simple instance when no arguments are passed in.

Waldemar agrees that there is a bug and will fix it.

Changes to Lexer for Characters in the Supplementary Range

The working group discussed possible options for (5) implementing lexer changes to support identifiers with characters in the supplementary range.

Markus proposes a two pass parser similar to the regular expression parser. The first pass would process the sequence of code units for the end of the identifier using simplified rules and emit a sequence of code points including unescaping any \u and \U sequences. The second pass would take the code units to generate the identifier.

Waldemar states that the lexer would still need to examine one or two code units to determine if the end of the identifier is reached. In addition, taking out some or all escape codes in the first pass wouldn't work because a \u000D is different from a <CR> in the source text, etc.

Waldemar has added char16 and char21 classes to handle code units and code points. He intends to handle case conversions by always converting to code points first.

There is agreement that supporting supplementary characters in strings will be straightforward.

Document Status Update

Waldemar has made the following updates:

(32) Modify algorithms to use "propagates call to every nonterminal expansion" wording where possible except for the case where values are returned
Done

(35) Should length be float64 instead of ulong?
Fixed. Array limit is now stored as a float64 and the array length limit is defined to be implementation dependent.

(18) Implement Array methods

Work item in progress

(26) Implement Error classes
(15) Implement toExponential method
(16) Implement toPrecision method
Completed

Waldemar is thinking about how to handle eval, function constructor, and programs composed of multiple pieces. Eval is tricky because of the various environments code can execute in.

Request for toCharArray

(39) procedure to convert string to array of UTF8 or UTF16 integer values is a proposal submitted by Joseph Myers (jmyers@lilly.csoft.net or joseph@myerskids.com) on the newsgroup.

The working group agrees that it this feature could be implemented relatively easily and that the use cases are valid. However, it is functionality that could be implemented in script and thus is not a showstopper. Given the fact that we are on the final stages for E4, the working group agrees that we will not accept this feature and consider it for a future version of the specification.