

## 2003 May 8 TC39 TG1 E4 Meeting

<b>Date</b>	March 8, 2003 10:00 AM – 5:00 PM
<b>Location</b>	Microsoft Corporation Building 41/1719 One Microsoft Way Redmond, WA 98052
<b>Convener</b>	Rok Yu (Microsoft)
<b>Editor</b>	Waldemar Horwat (Netscape)
<b>Participants</b>	John Schneider (BEA/AgileDelta) Rok Yu (Microsoft) Waldemar Horwat (Netscape) Markus Scherer (IBM)

### Agenda

The agenda was adopted as written.

### Document Review Process and Schedule

The document review process was accepted as proposed by Rok in email.

The review pipeline will be as follows:

1. One week before a scheduled meeting, Waldemar will send out updated draft w/ revision marks and list of sections to be introduced at the meeting.
2. Members will use the week to become familiar with the algorithms and evaluate if there are any major issues.
3. At the meeting, Waldemar will present the algorithms and answer any questions. We will discuss any major issues here.
4. Members will have until the next meeting to do a detailed review and get any minor issues resolved via email.
5. At the next meeting, we will formally accept the sections.

The status spreadsheet will be sent out before and after each meeting after the draft is sent out with the following codes:

- NA - not applicable
- Not ready - draft not yet completed
- Ready - ready to be reviewed, but not at coming meeting
- On deck - to be presented at coming meeting
- In discussion - opened at a previous meeting, reviews led to issues that need to be worked out before text can be finalized.
- In review - opened at a previous meeting, no major issues, wording in draft to be finalized for next meeting
- Accepted - agreed on final text

In order to have enough time to do test passes on the reference implementation, the working group has agreed that the algorithms have priority and will be completed before the prose. In addition, the sign off on the algorithms and the prose will be tracked independently. Sections 1 through 9 are prose only, and sections 10 through appendix A.4 contain both prose and algorithms.

Members have agreed to move to a biweekly meeting schedule, alternating between conference calls and face-to-face meetings. The following meetings have been scheduled. Unless otherwise noted:

- First day of face-to-face meetings is for E4, second for E4X
- Face-to-face meetings are from 10:00 AM – 5:00 PM on the first day to allow for morning travel, and 9:30 AM – 5:00 PM on subsequent days.
- Conference calls are from 10:00 AM – 12:00 PM
- Times are in PDT

Date	Venue	Notes
May 22-23	Conference call	
June 5-6	Meeting at Netscape	
June 19-20	Conference call	
July 1-2	Meeting at Microsoft	Moved earlier to accommodate July 4 <sup>th</sup> long weekend.
July 16-17	Meeting at Netscape	End of July meeting face-to-face meeting moved here because original dates conflicted with member schedules.
July 25	Conference call from 10:00 AM – 11:00 AM	Meeting to decide which specifications TG1 will submit for September ratification.

TG1 must submit the set of specifications that we plan for ratification at the September TC39 meeting by the end of July. The working group agrees that that it would be prudent to keep a one month buffer in the schedule before the September TC39 meeting to allow for unexpected complications. Given these dates, the working group has established the following checkpoints to track our progress.

- June 5 – reference implementation and algorithms complete
- July 24 – algorithm review complete
- second week of August – prose complete
- first week September – prose review complete

## Unicode and I18N

Markus led us through the Unicode and I18N proposals that he submitted to the reflector and posted to the ECMA site at:

- Unicode - <http://www.ecmadoc.net/docfiles/TC39-g1/2003/NOTPDF/2003tg1-001.html>
- I18N - <http://www.ecmadoc.net/docfiles/TC39-g1/2003/NOTPDF/2003tg1-002.html>

For each item, see corresponding proposal for more details.

### Unicode Version [Unicode proposal]

**Proposal:** To require implementations to use Unicode version 3.0 or higher.

The proposal was accepted.

Rok wanted to know rationale for updating the minimum bar to 3.0 rather than keeping at 2.1 or moving to a later version 3.1, 3.2, or latest 4.0 standards. Primary reasons are as follows:

Why 2.1 is not sufficient:

- Unicode 3.0 adds many characters in 2.1. Of specific significance is that it includes characters from the GB18030 standard which is required by Chinese government.

Why 3.1 or higher versions is too stringent:

- Mainstream platforms on which ECMAScript implementations run have not yet been updated to these later versions. We do not want to force ECMAScript implementations to carry around Unicode tables.

## Action Items

- Waldemar to ensure “2 Conformance” states that Unicode Edition 3.0 or later is required.

### **Ignorable Characters** [Unicode proposal]

**Proposal:** To change the set of ignorables from Cf to Default\_Ignorable\_Code\_Point, which is a superset of Cf.

The proposal was rejected.

After closer look at some of the additional characters included in Default\_Ignorable\_Code\_Point, Markus indicated he would withdraw this proposal.

Waldemar adds that unpaired surrogate code points are handled by UTF-16 requirement described in the “6 Source Text”.

### **Line Breaks** [Unicode proposal]

**Proposal:** To add the ISO Control U+0085 Next Line (NEL) to the list of LineTerminator characters.

The proposal was accepted.

Markus states that ISO Control U+0085 Next Line (NEL) is the Unicode character that is used to map EBCDIC next line character which is commonly used on mainframes.

### **Action Items**

- Waldemar to add 0x0085 to be added to “7.3 Line Breaks”

### **Identifiers** [Unicode proposal]

**Proposal:** To include the new Unicode property ID\_Start into the definition of ECMAScript identifiers.

The proposal was accepted

With the adoption of Unicode 3 as the minimum requirement, in addition to the use of ID\_Start, Markus suggests we take advantage of other Unicode classes introduced in Edition 3 and later.

In addition to character classes for identifier start and identifier continuing character, Markus suggests that we use White\_Space to define “7.2 White space”. In this case white space would be defined as White\_Space minus LineTerminators in “7.3 Line Breaks” and we should reorder the two sections.

Waldemar mentions that the character U+200B was part of category Zs used in E3, but is not in the White\_Space character class. The working group agrees to treat it as a non-white-space and consider it's inclusion as white space in E3 as a bug.

Even though ID\_Start is defined in Unicode Edition 4, with the inclusion of Other\_ID\_Start for backward compatibility, Markus suggests we use it and explicitly list the 4 characters in the Unicode 4 Other\_ID\_Start set for implementations that use Unicode versions before Edition 4.

### **Action Items**

- Markus to submit draft for “7.2 White space” based on Unicode White\_Space character class.
- Markus to submit draft for “7.5 Keywords and Identifier” to be based on ID start and continuing character classes, to use ID\_Start, and to add specifics to make implementations which conform to Edition 3.x behavior to match Edition 4 behavior.

- Markus to submit draft reversing order of sections “7.2 White space” and “7.3 Line Breaks” to allow the former to be written with Unicode character classes minus the characters in Line Breaks.
- Waldemar to incorporate Markus’ drafts into specification.

### **Escape sequence** [Unicode proposal]

**Proposal:** To add escape sequences to ECMAScript that can directly represent supplementary Unicode code points ( $\leq U+10FFFF$ ).

The proposal was accepted

Markus suggested several alternatives. The working group agreed on `\Uhhhhhhh` syntax which matches syntax that was recently added to C/C++/Python standards.

How `\U` affects regular expressions is still an open issue.

### **Action Items**

- Waldemar to modify “6 Source Text” to allow for extended `\U` syntax in strings and identifiers. Finalizing text is blocked on decision whether regular expressions will be updated to allow supplementary code points.
- Waldemar to modify “7.8 String Literals” to allow for extended `\U` syntax.

### **Terminology** [Unicode proposal]

**Proposal:** To update Unicode-related text in the ECMAScript standard to reflect changes in Unicode terminology. This would be best done editing the chapters with change bars for review.

The proposal was accepted.

In particular, Markus proposes that all relevant instances of code point should be replaced with code unit and all instances of character should be replaced with code point. Unicode uses code unit to identify a word in the byte sequence in an encoding of a string, and code point to refer to the abstract linguistic or typographical nit represented by a single scalar value from 0 through  $0x10FFFF$ .

### **Action Items**

- Waldemar to update all relevant uses of “code point” to “code unit”, and “character” to “code point”.

### **String Construction from Supplementary Characters** [I18N proposal]

**Proposal:** To extend `String.fromCharCode()` to accept supplementary Unicode code points. Instead of `ToUint16`, each argument would be converted with `ToInt32` (or `ToUint32`). Values `0..ffff` are processed as before. Values `10000..10fff` yield a pair of surrogate code units. Other values throw an exception.

The proposal was accepted.

Rok thinks that the `\U` syntax and this feature are strongly correlated. The fact we accepted `\U` is a strong reason fro accepting this feature.

The working group agrees that the algorithm will be modified to throw an exception on out of range Unicode code points (current behavior truncates to 16bit Unicode code units).

### **Action Items**

- Waldemar to write `stringFromCharCode` to account for proposal.

### **Wording of localeCompare** [I18N proposal]

**Proposal:** To update the text for this function by largely referring to relevant sections of the Unicode Standard, without expanding the required semantics of the function.

The proposal was rejected.

The working group agrees that the existing wording in E3 is confusing and must be changed in E4.

Waldemar and Rok point out that the intent of these sections in E3 was to allow implementation dependent behavior for these functions so that implementations on platforms without support weren't required to carry the tables around.

The working group agrees that we want to continue allow this freedom and reject the proposal to require localeCompare to treat strings that are canonically equivalent according to the Unicode standard as identical. To clear up the wording, the working group agrees to change the wording so that the normative part clearly indicates the behavior is implementation dependent, and that the suggested semantics is non-normative.

#### **Action Items**

- Waldemar to split out test so normative part specifies that behavior is implementation dependent and suggested semantics is non-normative.

### **Message Formatting** [I18N proposal]

**Proposal:** To add a function like `String.formatMessage(format, valueMap)` that supports variable-position inserts.

The proposal was rejected.

Rok states that given the current tight schedule and the fact that the functionality suggested is relatively easy to implement in script, this feature is not something that meets the bar.

The working group agrees, that we can E4 can ship without this feature.

### **Date/Time Formatting** [I18N proposal]

**Proposal:** To add functions to the Date object to get each month and day name ("März", "Dienstag"), to get localized AM/PM names, and to get date, time, and date+time format patterns as used by the Date object (e.g., an array like ["Y", "M", "D", "h", "m"]).

The decision on the proposal is postponed.

The working group is unclear as to the use case for the feature. Waldemar and Rok state that for the case described, HTML users currently use separate HTML for each of the locales, and handle ordering by rearranging HTML tags – not using script. This seems like a reasonable and workable solution.

Markus states proposal comes from his colleague and he will work with his colleague to understand the rational better.

#### **Action Items**

- Markus to review proposal with colleague to understand use cases for this feature.

## **Locale-Specific Functions** [I18N proposal]

**Proposal:** To review such functions and decide if a locale ID parameter or similar should be added for Edition 4.

The proposal is rejected.

The working group agrees that this is a relatively complex feature isn't a showstopper for E4 and given the current schedule, it cannot fit into the schedule.

## **Regular Expressions** [I18N proposal]

**Proposal:** To extend Regular Expressions (CharSet etc.) to support supplementary code points.

The decision on the proposal is postponed.

The working group is concerned that the set of changes required to support this is too large to be executed in the remaining schedule.

Markus believes the feature will not take significant amounts of work.

The working group agrees to postpone the decision until Markus can write up the modifications needed to regular expressions

### **Action Items**

- Markus to make modifications to regular expression definitions to support supplementary code points.

## **Algorithm Reviews**

### **Review Order**

Waldemar states that sections 1 through 10 contain definitions and internal algorithms and suggests that we should review them on demand as their use is encountered in other algorithms.

The working group agrees that this review order is an effective way to cover the material.

### **Criteria for Deciding on Features**

The question was raised as to which criteria the working group will use to decide which work items we agree to. i.e. Does it make sense to make I18N changes but not keep import?

John states that the intent of the prioritized list of features that he presented last face-to-face meeting was to try and make clear the priorities for making such decisions.

Rok states that he thinks that given the work that needs to be completed by September, it's already too late to be adding any features except for the show stoppers. Any features we add past June code complete date has high probability of pushing the schedule past the deadline.

Waldemar states that spending time defining a bar is moot because the goal is to be algorithm complete by the next face to face meeting in June.

Rok agrees to postpone the discussion on establishing decision criteria, but qualifies this position by explicitly stating that the agreement then is what is completed in June is what's in, and what's not, is not. The bar for adding features after that point will be very, very high. Rok adds that getting the object model algorithms from E3 ported to E4 are higher priority than adding new features for E4 such as I18N and import and Waldemar should work on completing them first.

The working group agrees to these principles, prioritization, and process.

### **Changes from Previous Version**

John notes that the import of packages has been cut from this version of the spec. He mentions this because import was on his short of list features that are important to BEA and he did not recall ever agreeing to cut this.

Rok concurs that we did not agree to cut the import of packages, but only the declaration of packages.

Waldemar states he was unclear on this point and thought the agreement was to remove the entire feature because the easiest way to add package importing and define what it does is to add package definition. He believes that package definition is simpler to add than package importing and without the definition part, it's hard to state what import does. Waldemar thinks he can add back the import statement fairly easily.

John notes that "9 Definition of namespace" does not contain a URL as agreed to in a previous meeting.

Waldemar states he didn't get to the action item and his intent is to complete this work item for the June meeting. Furthermore, he intends to define the algorithm so that new Namespace("") will return the public namespace.

### **Notes on Algorithms**

#### **Sections Covered**

Waldemar walked us through sections 11 through 12.4 along with definitions that were used and defined in sections 9.1.9, 9.1.10, 10.10, and 10.13.

#### **11 Evaluation**

Waldemar states that this section is somewhat redundant with "16 Programs". and is considering either moving section 16 forward, or keeping this section as prose only. It is not ready for acceptance.

#### **High Level Algorithm Structure**

The algorithms have a 3 pass evaluation.

- Validate
- Setup
- Eval

Compilation consists of Validate, Setup, and Eval with phase == compile. Execution is via Eval with phase == Run.

Validate recursively visits all the nodes in the semantic tree established by the grammar to:

- established chain of static environments for every point in the program
- established a compile time context for every point in the program
- handle any syntax errors
- resolves attributes
- populates properties

Setup recursively visits all the nodes in the semantic tree established by the grammar to determine the type of all variables and expressions.

Eval with phase == compile is used to evaluate compile time constants.

### **Action Items**

- Waldemar to remove exclude and include from sections 12.1 & 7.5.
- Members to review carefully reviews sections 9.1.9, 9.1.10, 10.10, 10.13, 12 – 12.4 for acceptance at next face to face meeting. Issues to be discussed in email as they are encountered.
- Members to do first pass on 12.5 – 12.21. Waldemar to be walking through these sections during the conference call.