

**Minutes of the:** **19<sup>th</sup> meeting of Ecma TC39**  
**held in:** **Cupertino, CA, USA**  
**on:** **16 – 18 November 2010**

## 1 Opening, welcome and roll call

### 1.1 Opening of the meeting (Mr. Neumann)

The meeting (hosted by Apple at their premises in Cupertino, CA) was opened by **Mr. Neumann**, Chair of TC39 at approximately 10:00 AM on 16<sup>th</sup> November 2010 ([2010/058](#): Venue for the 19th meeting of TC39, Cupertino, November 2010).

### 1.2 Those in Attendance included:

John Neumann	Ecma International
Istvan Sebestyen	Ecma International
Allen Wirfs-Brock	Microsoft
Luke Hoban	Microsoft
Cormac Flanagan	UCSC
Rob Sayre	Mozilla
Cameron McCormack	Mozilla
Mihai Nita	Adobe
Paul Cotton	Microsoft
Sam Ruby	IBM
Adrian Bateman	Microsoft
David Fugate	Microsoft
Shawn Steele	Microsoft
Chris Sells	Microsoft
Nebnojsa Ciric	Google
Dave Herman	Mozilla
Tom Van Cutsem	University of Brussels (phone – part time)
Mark Miller	Google
Douglas Crockford	Yahoo!
Waldemar Horwat	Google
Alex Russell	Google
Erik Arvidsson	Google
Oliver Hunt	Apple
Sam Weinig	Apple
Brendan Eich	Mozilla
Jungshik Shin	Google
Peter Constable	Microsoft

Philippe le Hageret                      W3C (phone – part time)

### 1.3 Host facilities, local logistics

**Oliver Hunt** outlined the services for Lunch. **Mr. Sebestyen** has invited TC39 for a Dinner on Wednesday evening.

**Mr. Horwat** agreed to take technical notes for the meeting. The technical notes on the Internationalization were taken by **Mr. Ciric**.

## 2 Adoption of the agenda ([2010/059 rev. 2](#))

The agenda has been approved by the meeting.

## 3 Approval of Minutes from September ([2010/057](#))

The minutes from the September TC39 meeting were approved without modification.

## 4 Report from the Secretariat

### 4.1 Internationalization standard proposal (Google) [2010/052-Rev2](#)

The meeting agreed to hold a break-out that was held by a group of interested TC39 at the premises of Microsoft. **Mr. Ciric** has provided a report on the meeting that is included in the Annex of the Technical Notes..

#### 4.1.1 Terms of Reference

#### 4.1.2 Development plan and schedule

#### 4.1.3 Ad hoc members and chair

### 4.2 Progression of ES 5

The meeting discussed the possible way forward to conclude the JTC 1 fast-track of ECMA-262 Edition 5. If needed the BRM is planned for Wednesday, January 19, 2011 at 15:00 Pacific Time (that equals to Thursday, January 20, 2011 08:00 Tokyo Time). That meeting may not happen, if everything agrees with the submitted Ecma documents.

For this the following steps have to be taken:

\* The revised document and Disposition of Comments to SC 22 will be sent to the SC 22 Secretariat by the end of November 2010. This must be circulated 4 weeks prior to the BRM so late December would be latest. *Secretary note: This was actually done on November 18, 2010.*

\* The BRM will be scheduled in the San Jose/Sunnyvale on January 19, 2011 and will be mixed onsite/conference meeting. SC 22 needs to send out a meeting notice and draft agenda no later than mid December 2010.

\* **Mr. Istvan Sebestyen** will chair the meeting, **Mr. Rex Jaeschke** SC 22 will also participate in the meeting.

\* At the conclusion of the BRM, we will see if Japan changes its votes from “No” to “yes”. In that case but no longer than one month after the close of the BRM, SC 22 would distribute a new version of the document that includes all changes agreed to at the BRM as well as a disposition of comments report approved at the BRM. Then the standard would be published by JTC1. The document should be submitted for either FDIS or 2nd DIS ballot (if submitted technical comments result in substantive changes and if there is no 100% agreement in the BRM).

It is not clear if SC 22 will require a BRM. This will depend very much how the final documents by the Editor are accepted by SC 22, and especially the Japanese NB.

#### 4.2.1 Editor's Proposed Disposition of Comments Report on 16262 Edition 5 (2010/055)

**Allen Wirfs-Brock** has reported about the work he did as Editor. The work is done. 60-61 zip files. One disposition of comments, the other is the new documents. TC39 has accepted the Disposition of comments.

#### 4.2.2 Ecma editors proposed new draft for submission to SC 22 for BRM

A new draft standard was approved by TC39. The Ecma Secretariat will submit both documents asap.

Then Ecma TC39 has to work on EcmaScript 5.1. This will include a new annex with the deltas between 5.0 and 5.1. The Editor will send this out by the end of December 2010. The goal is that the Ecma GA will approve this new Edition in June 2011. The Ecma Secretariat will also publish the draft on the Ecma website and publication on the wiki will also be done. It was agreed that on the web pre-published version the word "draft" should be put on.

TC39 expressed its gratitude to the Editor, **Allen Wirfs-Brock** for his hard and excellent work.

On November 18, 2010 the following documents have been sent by the Ecma secretariat to the SC22 Secretariat:

- The Word file is the Editor's Disposition of Comments document that describes the resolution for all the comments submitted by the ISO members
- The zip file contains two copies of the document. The "with markup" copy contains full change markup relative to the original version submitted to ISO. The "clean" copy is the same content with all changes accepted and no visible markup.

### 4.3 Progression of Test Technical Report and EcmaScript.org

#### 4.3.1 Review CC Meeting re: Test 262

**Mr. Neumann** gave a report about the presentation of the testing webpage run by MS in the CC. The CC has approved the concept for such testing.

The work on the Technical Report regarding ES 5 testing is continuing.

#### 4.3.2 Status of contributions

Microsoft and Google should send the software copyright contributors agreement to Ecma. Microsoft claimed that this was already done. **Mr. Sebestyen** will check it if this was received by the Ecma Secretariat. It should have arrived via normal mail.

This has to be cleaned by the next meeting of TC39.

#### 4.3.3 Test development plan and schedule

#### 4.3.4 Prototype website

<http://es5test.ep.interop.msftlabs.com/ecmascripthome.html>

<http://es5test.ep.interop.msftlabs.com>

### 4.4 TC39 possible relationship with Khronos (WebGL WG and Typed Arrays)

### 4.5 W3C Joint work items

The Liaison relationship between Ecma International – W3C, and TC39 – W3C Technical Groups has been discussed.

**Mr. Sebestyen** has reported that on the "higher level" Ecma International – W3C no such recognized relationship exists. He pointed to a large table on the W3C website that includes all W3C liaison, and where Ecma international is not mentioned.

He said that from PR point of view such an entry would be useful, and he also added that he has contacted the W3C Liaison Committee how to achieve that. Members of the Liaison

committee did not expect any difficulty with that, but it takes some time to process it within the W3C. They have asked to provide the necessary information to fill in the W3C matrix (e.g. Ecma Liaison Representatives and common subjects). It was decided that as Ecma Liaison on that list **Mr. Sebestyen** and **Mr. Neumann** should be included. As interesting W3C groups for liaison the HTML, Web Applications, Internationalization and Security have been agreed to. **Mr. Sebestyen** will communicate the above to the W3C Liaison Committee. This was done.

Regarding liaison between TC39 and the respective W3C groups it was confirmed by the meeting that the current loose style of co-operation should continue, e.g. mutual information when needed, joint meetings case by case, etc.

#### 4.6 Report from the CC meeting

**Mr. Sebestyen** gave a brief report about the recent CC meeting, especially about those points that were interesting for TC39.

He reported that the CC was pleased with the progress of TC39.

He CC has agreed that after the JTC 1 approval of the ECMA-262 Edition 5 fast-track, harmonization between the ISO/IEC and Ecma version of the standard harmonization is needed. Such a harmonization is expected for approval at the June 2011 GA. The CC has showed understanding that the new Edition of ECMA-262 should be "Edition 5.1".

## 5 WebIDL Discussions

### 5.1 Topics from Cameron McCormack

Modules ==

<http://dev.w3.org/2006/webapi/WebIDL/#idl-modules>

Static members ==

<http://dev.w3.org/2006/webapi/WebIDL/#idl-interfaces>

Operation overloading ==

<http://dev.w3.org/2006/webapi/WebIDL/#idl-operations>

Special operations ==

<http://dev.w3.org/2006/webapi/WebIDL/#idl-special-operations>

Sequences and arrays ==

<http://dev.w3.org/2006/webapi/WebIDL/#idl-sequence>

<http://dev.w3.org/2006/webapi/WebIDL/#idl-array>

<http://dev.w3.org/2006/webapi/WebIDL/#es-sequence>

<http://dev.w3.org/2006/webapi/WebIDL/#es-array>

Interface types ==

<http://dev.w3.org/2006/webapi/WebIDL/#es-nullable-type>

Deciding what to do with functions passed too many/few arguments ==

<http://dev.w3.org/2006/webapi/WebIDL/#es-interface-call>

Managing multiple inheritance and mixin interfaces ==  
<http://dev.w3.org/2006/webapi/WebIDL/#idl-interfaces>

DOM exceptions inheriting from Error ==  
[http://www.w3.org/Bugs/Public/show\\_bug.cgi?id=10622](http://www.w3.org/Bugs/Public/show_bug.cgi?id=10622)

Property enumeration order ==  
[http://www.w3.org/Bugs/Public/show\\_bug.cgi?id=11033](http://www.w3.org/Bugs/Public/show_bug.cgi?id=11033)  
[https://bugzilla.mozilla.org/show\\_bug.cgi?id=524423](https://bugzilla.mozilla.org/show_bug.cgi?id=524423)

## 5.2 Topic from Allen Wirfs-Brock

Refactoring "ECMAScript binding" into separate language binding and Browser extended ECMAScript object semantics sub-specifications.

## 5.3 Topic from Dave Herman

Modules

## 5.4 Topics from Minutes from September meeting

- 5.4.1 Relationship between ECMA-262, WebIDL, host object semantics, and the JavaScript binding for WebIDL - how would we present our preferred organisation of the WebIDL document?
- 5.4.2 Features of WebIDL that are supported for historic compatibility but strongly discouraged from future APIs
- 5.4.3 Roadmaps for TC-39 and W3C Web APIs - get everyone on the same page
- 5.4.4 Identifying shared goals and understanding how the groups coordinate their work towards those goals

## 6 Discussion of ES harmony (technical contributions are available and can be found on the ES wiki)

- 6.1 **const functions extended with joining**  
<[http://wiki.ecmascript.org/doku.php?id=strawman:const\\_functions](http://wiki.ecmascript.org/doku.php?id=strawman:const_functions)>.
- 6.2 **egal** <<http://wiki.ecmascript.org/doku.php?id=strawman:egal>>
- 6.3 **iterators**  
<http://wiki.ecmascript.org/doku.php?id=strawman:iterators>
- 6.4 **additional "instanceof" trap for proxies**  
<http://wiki.ecmascript.org/doku.php?id=harmony:proxies>
- 6.5 **binary data specs**  
[http://wiki.ecmascript.org/doku.php?id=strawman:binary\\_data](http://wiki.ecmascript.org/doku.php?id=strawman:binary_data)

## 7 Date and place of the next meeting(s)

18 – 20 January 2011, Hosted by Yahoo! In Sunnyvale, CA

## 8 Closure

**Mr. Neumann** has thanked Apple for hosting the meeting and Ecma International for hosting the fine dinner on Wednesday evening. He also thanked the experts for the good and



constructive discussions and the good progress made at the meeting. The meeting was closed at 16:00 local time.

=====

**Annex: Technical Notes by Waldemar Horwat:**

Subj: Rough notes for day 1

Here are my very rough notes for today's IDL discussions. In many cases I couldn't tell what was actually decided rather than merely discussed.

Waldemar

-----

IDL day: Lots of discussions, exploring various points and making suggestions. Mostly exploratory -- not in the proper frame today to reach significant concrete conclusions.

Allen: WebIDL encouraged specification of APIs that are awkward or detrimental in ECMAScript

Alex: Bad API example: DOM collections

Allen: Legacy features should be segregated and labeled "legacy" and not used to design new APIs. Specification feels like the APIs were designed for Java and then adapted to JavaScript.

Points from this meeting are "deja vu" repeats of the points from the 2009 November meeting.

Oliver: IDL mappings onto ECMAScript details like prototypes are underspecified.

Allen: Overload resolution should not be used as an extension of the call internal method. Instead, how the function behaves with arguments of various types should be specified.

If ECMAScript is the primary target, why have multiple inheritance? It's for SVG. Debated the usefulness and appropriateness of multiple inheritance in WebIDL.

Lunch discussion about the merits of adding a cleaner array type.

Some implementations currently cheat on ES5 semantics on arrays, failing to run a setter in a prototype when writing into a hole in an array. Mozilla detects this case and managed to do it right.

Debate over specification of rejection of mistyped arguments by algorithms (as in ES5) or by implicit type-validation rules together with an IDL type signature.

WebIDL: There are only a limited number of places where a native object may be used.

Discussion of annotation of ownership transfers across APIs in the context of synchronous callbacks or retention of references by the callee in some global data structure. When is it safe to pass mutable state?

Presentation request: Wish that specification of WebIDL were linked to real API examples where those IDL features were used.

What are the problems with making NodeList inherit from Array?

Interface and mixin discussion.

Brendan: If WebIDL interfaces are abstract, they shouldn't be reflected into ECMAScript. (If they were, instanceof would be problematic because the ECMAScript prototype chain doesn't support mixin multiple inheritance.)

What should "new NodeList" do?

Conclusion: Abstract interfaces are not reflected into ECMAScript.

```
interface Node
```

```
interface EventTarget
```

```
Node implements EventTarget
```

```
Interface Element: Node
```

Prototype chain of an instance of Element is:



Element.prototype

Node.prototype

Object.prototype

MarkM: Do we really need abstract interfaces inheriting from abstract interfaces? This is complicating the spec-to-ECMAScript-API translation.

Those are useful in cases such as DOM 1 nodes vs. DOM 2 nodes. Debated.

Need good names for how to refer to reified vs. spec sugar interfaces.

Allen: Move all IDL places that can't be expressed natively in ECMAScript into a separate section of the spec.

We agree that host object extensions should be as small as possible.

Discussed the merits of overloading vs. union types.

For an overloaded method that takes instances of one of two different interfaces, passing null will satisfy both interfaces. In this case IDL says that both overloads of the method should do the same thing with null.

If you define a parameter with a union type I1|I2 where I1 and I2 are abstract interface types, the callee has no way to distinguish whether the argument is a member of I1 or I2 because abstract interface types are not reified. Thus the caller can only invoke methods that are common to I1 and I2.

Do we need unions of abstract interface types?

Waldemar: IDL "getter" and "setter" are confusing. Defining interface I {  
 getter float foo(...) ...  
}

will define every every property on instances of I to be a getter \*except\* for the foo property, which is a regular method.

53-bit int type? Current IDL long long type is hackish -- it allows loss of precision beyond  $2^{53}$ . On the other hand, there is a proposal to introduce high-precision integers into ECMAScript, so 53-bit ints seems like another hack.

---

es-discuss mailing list

es-discuss@mozilla.org

<https://mail.mozilla.org/listinfo/es-discuss>

=====

Day2 Waldemar's notes:

From: waldemar@google.com

To: es-discuss@mozilla.org

Sent: 11/17/2010 8:23:30 P.M. Eastern Standard Time

Subj: Nov 17 meeting notes

WebIDL: Can abstract interfaces have static members? Don't see why not -- they'd just be spec sugar for adding the same static member to concrete classes that derive from those abstract interfaces. As usual, it would be a spec error to have a collision.

Debate over combination of overloaded methods from different base abstract interfaces. Issues come up with combining overloads of abstract types -- per yesterday, the structural "union" of two abstract interfaces A1 and A2 can contain instances that are in neither A1 nor A2. General feeling is to avoid the issues if feasible.

Property enumeration order: Decided that this is a TC39/ECMAScript issue, not a WebIDL issue.

Dealing with argument count mismatches: Ignoring additional arguments is useful for upwards compatibility (example: adding an extra dirty region argument to draw methods that would be ignored by older browsers that would just redraw the whole canvas).

Overloading is a mismatch with future-proofing argument count mismatches. Proposal: treat each function call as having infinitely many trailing "undefined" arguments, and overload only on types, not argument count.

Brendan on special operations: "don't want to see any more of these darken our door". Trouble is that these kinds of catch-alls allow most names but not all names, leading to brittle or exploitable code. It's better to provide get and set methods.

However, Stringifiers are ok.

Error objects have their string properties (name and message) defined on their prototypes in ES3. IDL errors can create instances with instance properties for name and message rather than delegating to a prototype, and everything ought to work.

Browsers like to add other properties to Error objects they create. Not clear how to link into that functionality. Throw this back to TC39's core language discussions.

Returned sequence types are returned as arrays. That means that they must be copied each time they're returned. Should they be frozen? Not much enthusiasm for that....

What about passing arrays into IDL? The IDL can just access whatever it wants because it has control until it returns. Except that it doesn't if there are getters, setters, or proxies in the data structure passed to IDL -- the order of accesses is discoverable, and the data structure can mutate itself as it's being read.

The proposal of having IDL read data structure "as if copied" is not practical. Some methods might only want to access one element of an array and don't want to copy the whole thing.

We won't require users to freeze arrays before passing them into IDL. That would be too cumbersome.

We won't require IDL to freeze arrays before passing them to users.  
Every array will be fresh.

Discussion about indexGetter and indexSetter. Separate path for looking up numerical indices?

John, Brendan: Want to avoid high management overhead for cooperating with W3C. A lot of liaisioning formality or large meeting would be undesirable.

Discussion about desirability of writing a style guide or joining TAG. Style guide may be ineffectual -- it's better to just have someone who understands style review proposals -- while TAG has no teeth to enforce its mission, so it's practically been ignored.

---

Next meeting on Jan 18-20 at Yahoo. Ballot resolution meeting at 3pm on Jan 19.

We'll send the final ISO draft to ISO tomorrow (Thursday) and also place the draft on the ECMA website. Allen: Need to re-designate the document as a draft if we let it out now.

Consensus on removing the test results altogether from the test262.ecmascript.org web site. Having results there would provide too many incentives for trying to game the system rather than build a good test suite.

Debate on whether do commit-before-review or review-before-commit on the test suite.

Would be good to get rid of the powershell platform dependency when submitting tests.

Discussion about whether we can get rid of -0. Some are in favor but no, we can't. We also can't make -0 !== +0. Either would be a large breaking change.

Some desire to make the identity operation syntactically as attractive as `===`. Could we make the identity operation into an infix operator named "eq"?

Waldemar: Syntactically, we could, and we wouldn't even need to make "eq" into a reserved word. The production would have to be:

```
expr1 [no line break here] eq expr2
```

The reason the [no line break here] has to be in the first gap instead of the second one is to maintain compatibility with the existing code that counts on semicolon insertion:

```
a = x
```

```
eq = y
```

which would continue to parse as:

```
a = x;
```

```
eq = y;
```

Agreed to move this into the proposal stage.

const function joining: Vigorous debate. Some don't like specifying the optimization algorithm. What is its asymptotic complexity?

Oliver would object if it's greater than  $n \cdot \log(n)$  but might object anyway.

Waldemar: Opposed to mandating making dead code change semantics of live code, as in:

```
const debug = false;
const divisible(m) {
  return const(list) {
    return list.filter(const(n) {
      if (debug) log(list); // dead code
      return n%m === 0;
    });
  };
}
```

Now the innermost function has different `===` function instance behavior thanks to the dead code it contains. The behavior ought to be identical to:

```
const divisible(m) {
```

```
return const(list) {  
  return list.filter(const(n) {  
    return n%m === 0;  
  });  
};  
}
```

Debate over syntax. No particular resolution.

Everyone seems to have a different view on this.

A: Skeptical about syntax.

B: Like it but would want to make joining implementation-defined.

C: Don't think users will want to freeze functions much.

D: Not convinced secure mashups have been demonstrated.

E: Would like even shorter syntax.

F: Figure out asymptotic complexity.

G: Use  $\lambda$  as the syntax.

---

es-discuss mailing list

es-discuss@mozilla.org

<https://mail.mozilla.org/listinfo/es-discuss>

=====

From: waldemar@google.com

To: es-discuss@mozilla.org

Sent: 11/18/2010 7:09:02 P.M. Eastern Standard Time

Subj: Nov 18 notes

Third day rough meeting notes.

Waldemar

instanceof trap for proxies: Withdrawn. Agreed to drop it for now.

Proxy default forwarding handler. Agreed that there should be an easy way to delegate to a default forwarding handler.

Waldemar: Why not use prototypes to do this?

MarkM, others: Would work, but need to write a proposal for it.

MarkM: Arguing for simplicity (understandability) of the modules proposal

Discussed the merits of modules, asynchrony, scoping, and evaluation phasing. Agreed that static resolution of bindings should produce results indistinguishable from later dynamic resolution of those bindings. Thus, for example it's not ok to asynchronously import \* into an outer lexical scope. Also need to ensure that asynchronously loading multiple scripts on a page doesn't violate this rule for the outer lexical scopes, perhaps by making the "global" scope no longer be reflected as an object.

DaveH: Would like to see Harmony lexically scoped all the way up.

Discussion of IEEE 754r decimal cohorts.

Debate over whether for-in should be conceptually restricted to iterating only keys or whether users should be able to do things like:  
for (var val = values(x)

Other points of contention:

Iterating keys of an array should return numbers or strings?

What should be the syntax for general iteration?

Provide a different syntax for user-defined iteration? for (var key : iterator)?

Provide a way to iterate only the shallow enumerable properties of an object?

Consensus that we should have iterators.

Generators: Need to have a distinctive syntax in the preamble to distinguish them from functions.

Finally inside generators is also an issue because it might not run.

Waldemar: don't allow yield inside a try.

Binary data: We have array buffers (not on wiki) and the binary data spec on wiki. Array buffers allow accidental aliasing and endian dependence, and we'd like to not have them.

Two very distinct use cases:

WebGL arrays: endian-ignorant, fixed size, preallocated

File parsing and other such I/O: need to explicitly specify endianness, dynamically construct representation, deal with variable-sized items such as strings. Endianness is a property of the i/o routines, not of the underlying data buffer.

An arraybuffer is like a void\* -- it's a container for view data structures to index into, where those other data structures impose a structure on a slice of the arraybuffer.

DaveH: Would be good to enforce no aliasing of views into arraybuffers.

Debate over what primitives should be supported:

Waldemar: In-memory contiguous buffers that allow pointers to simple things such as strings. Example with three contiguous records:

```
int
int
char* -> "Hello"
int
int
char* -> "World"
int
int
char* -> "\n"
```

Each record is fixed-size and has a fixed contiguous structure but whose layout and exact byte size is otherwise invisible to the ECMAScript programmer. The records contain pointers to the strings, but the strings themselves are stored somewhere else on the heap. There should be no way to read/write these into a file except by explicitly describing how it's serialized or deserialized. Of course, all this could also be done using traditional ECMAScript arrays and objects but at a much higher cost when the data is things like graphics data.

DaveH: An in-memory array of bytes (like an arraybuffer) together with ways to map views onto it to interpret things like ints, strings, or offset-pointers in some external data format (such as a JPEG).



Waldemar's objections to providing views into an in-memory array of bytes:

- There is a huge menagerie of basic data types used in files: Many different kinds of string termination, many different kinds of offsets and record counts, computed lengths, etc.
- This structure is not appropriate for general modifications. An ECMAScript program cannot practically insert something into the middle without adjusting offsets, lengths, etc. in the block records around the inserted data. Instead, the offsets, lengths, etc. should be consumed by deserialization and generated by serialization.

Turns out that the GL interface doesn't let ECMAScript write directly into GPU memory. The methods make copies as needed.

More discussion of data buffers.

---

es-discuss mailing list

es-discuss@mozilla.org

<https://mail.mozilla.org/listinfo/es-discuss>

=====

**Notes taken by Nebojsa Ciric on**

**Internationalization project discussion: Day 1 MS Campus:**

Tentative list of items we discussed at the 1st i18n EcmaScript meeting, 16th of November 2010.

Participants:

Jungshik Shin (Google), Mihai Nita (Adobe), Peter Constable (Microsoft), Shawn Steele (Microsoft), Mark Davis (Google) and Nebojsa Ciric (Google).

The group agreed on chair and editor for the project.

Chair is **Nebojsa Ciric** and editor is **Peter Constable**.

Scope of the proposal was refined:

- Timezones - mismatch Olson and Windows TZIds - potentially out of scope (v1.0) - Microsoft to review mappings, possible problems on other platforms too (POSIX?)
- date/time formatting/parsing
  - patterns/skeletons/enums - discuss
  - month and day names (get the actual name of month/name) - if we don't have skeleton/pattern then we need separate method to get them
  - support time

- not in v1.0 - duration, intervals (like 37:45h or Oct-Nov 2010)
- number/currency formatting/parsing
  - patterns vs. fields - perhaps subset of CLDR patterns
- collation
  - Platform dependant
  - Extensible set of parameters - need to discuss v1.0 set
  - Minimal set - ignore case/variants, ignore accents ~== strength
- message formatting
  - Plain string substitution - needs further specs
  - No plural/gender - not for v1.0 - needs deeper discussion
- message catalogs - discuss
- locale object
  - Language, Locale, Region - add docs/examples
  - We need to discuss subtle issue of en-US language vs US region - make API to make clear we are guessing the region if you pass en-US (no region info there, just language dialect).
- Users Preferred Locales List - v1.0
  - matching preferred locales to supported locales
- transforms - probably v1.0
- locale dependent case
- normalization
- extra data (ltr/rtl, units...) - need to discuss
- Error/fallback reporting
  - Ability to get actual parameters that were used for the operation back (say collation) – discuss

=====

Attendees,

please review the meeting notes (below) before Mr. Neumann adds them to the main meeting notes.

Elected chair - Nebojsa Ciric

Editor - Peter Constable

Participants - Peter Constable (Microsoft), Mark Davis (Google), Jungshik Shin (Google), Shawn Steele (Microsoft), Mihai Nita (Adobe)

Using wiki - we should be able to append new pages to the original document with bigger comments - else we append to the end of the document.

Make sure any code is under Ecma software policy. - We should ask John or Sebastian about that.

Should we have JavaScript library (native) as reference implementation? It's probably too much work, but we could discuss it more.

Role of editor - Peter is taking on editor role - reviews are done by TC39 too.

Mihai presented about ActionScript API and similarities/differences to our proposal. He also pointed out problems in cross platform development (POSIX, Mac, Windows, ICU...).

Discussion about Locales, and differences between platforms. Also en-US is a language not language and region. We should introduce notion of the region.

Method that returns user preferred list of languages. One can't access accept languages right now for example.

Matching locale id-s. Should match what we want with what we have.

Extra data - rtl/ltr, units - maybe expose interface to allow developers to get to this data. What to do with vendor specific data? Maybe prepend google.com to the key to make it unique (if it's google unique element).

Date/Time formats - enum/skeleton - skeleton looks like a good general solution but needs more discussion. Enums may be good enough as the must have.

Normalization of unicode strings - NFC, NFD

Add a reference to actionscript documentation to strawman.

Expecting variation in results across platforms. Collation for sure. We probably shouldn't standardize the results.

--

Nebojša Ćirić

=====

Internationalization will meet also in January 18, 2011.