Notes of the: **Meeting of Ecma TC39 ad hoc on Internationalization**

held on: **19 April 2013**

Location: Wikimedia, San Francisco, CA, USA

Attendees: Nebojša Ćirić (Google), Alolita Sharma (Wikimedia, invited), Norbert Lindenberg (Mozilla), Suresh Jayabalan (Microsoft), Eric Albright (Microsoft), Andrew Glass (Microsoft), Jungshik Shin (Google).

Minute takers: Alolita Sharma, Nebojša Ćirić, Suresh Jayabalan, Eric Albright, Jungshik Shin, Norbert Lindenberg.

Minute editor: Norbert Lindenberg.

1 Agenda:

- Status of ECMA-402, 1st edition, implementations and test suite.
- Status of ECMA-402, 2nd edition, draft specification.
- Discuss impact of ES6 slip to December 2014 on second edition of Internationalization API spec.
- Decide whether to define default for formatting of time zone name, and if yes, which one.
- Review current strawman proposals.

2 Minutes

2.1 Implementation of first edition

- Chrome: on desktop since v24 without 'V8' prefix.
- Opera: Adopting 'Chromium/Blink/V8'. Beta on mobile, Desktop soon.
- Firefox: Disabled now. Will be enabled in Firefox 23 in August.
- IE: working on it; have reference implementation.
- Safari: unknown.
- Amazon Kindle: old spec implemented, but the final standard not yet implemented.

2.2 Test suite for first edition

- Incomplete coverage (~15-20% coverage).
- More test cases needed.
- We were planning to publish Ecma technical report similar to the one on Test262, but can't do that until we have better coverage.
- Chrome: several failures - mostly because of being V8 extension without privileges (will try to address with V8 team). ICU issues (upgrading ICU to 50 or later will fix them).
- Wikimedia Foundation can contribute some test cases (pre-req: become member, sign contributor agreement, and get Mercurial access).
- Microsoft: can contribute – talk to Brian Terlson.

2.3 Specification of second edition

- Norbert published first draft end of February.
- Added full support for IANA time zone names in DateTimeFormat constructor.
- Added locales argument to String.prototype.toLocaleLowercase, toLocaleUpperCase. So far case conversion only per Unicode, but can therefore require support for all locales. CLDR 23 added locale-sensitive casings for Greek, Lithuanian etc, but if they haven't been added to Unicode, we can't require them, so we'll have to change to normal locale negotiation.

2.3.1 Time zones

Decide whether to define default for formatting of time zone name, and if yes, which one (similar to using ISO 4217 code for currencies if nothing else available. Decision: Don't define ultimate fallback.

Handling of UTC/GMT-like time zones: Decision: Converge all variations into UTC.

2.4 Schedule

Impact of ES6 slip to December 2014 on second edition of Internationalization API spec.

- ES6 schedule : Dec 2014, a big slip from Dec 2013.
- Internationalization v2 was planned to be released 6 months after ES6 (June 2014)
- Alternatives for v2: Stick to June 2014 based on ES5, release with ES6 in Dec 2014, stick to delta to ES6 and release in June 2015.
- ES6 benefits : Unicode support (non-BMP), iterators, modules.
- ES6 feature completeness targeted for Sept 2013, TC 39 has to approve by Sept 2014 so that Ecma GA can approve in Dec 2014.
- Decision: We'll align with ES6 in spec content and schedule, i.e., target Dec 2014 GA approval.

Related issue: Separate test repositories for ES5 and ES6 (how do you handle conflicts).

2.5 Message Formatting

http://wiki.ecmascript.org/doku.php?id=globalization:messageformatting

- Overview of JavaScript libraries implementation (different from string templates).
- Simple format (w/o plural and gender)
- Addresses ICU message formatting problem of escaping
- could combine SSF and modified ICU solution
- Microsoft does not have platform support for this; just private adhoc implementations
- Handling complex strings with gender/plural combinations
- How does this n by n matrix get interpreted for non-technical translators
- GENDER uses general SelectFormat class, parameters are passed, populated at run time
- PLURAL has different sentence forms for different numbers
- Complex use cases: (gender(plural)), (plural(gender))
- Need better examples for use cases

- Stringify JSON objects? (needs to be validated)
- Use cases for nesting
- Overall - JSON format should work
- Acceptance criteria: Whatever formats are defined have to be consumable by localization teams

Template strings need to be addressed; Justification for not including template strings and list of issues that need to be addressed in a blog entry for discussion list

Action item: Norbert and Nebojša will work on this justification.

Discussion of Norbert’s proposal to use toLocaleString with per-argument options provided to MessageFormat constructor

Action item: Nebojša will incorporate Norbert’s proposal/comments into the Message Formatting proposal.

2.6 Text Segmentation

http://wiki.ecmascript.org/doku.php?id=globalization:text_segmentation

Use cases:

- GMail: offline messages for indexing (word-break).
- Google docs: line breaking on the client-side.
- Counting words in a document.

Q1: Do we want to force our clients to breakup strings into whatever they use or do we use an abstract interface

Q2: Pass a string to segment to ctor or 'create once and adopt a string to segment'

Q3: add a new param to existing String.prototype.split ()?

Q4: numSegment() is expensive.

Should text segmentation be integrated into RegExp? UTR 18 Unicode Regular Expressions Level 3 suggests locale sensitive matching; however, this is not implemented by any implementation that we know of. Likely issues are interaction of locale sensitive matching with other functionality; performance; need for large tables and dictionaries.

Makes sense to have a separate API.

Action items (Rich):

- Describe use cases
- Use only segmentContaining and use iterators from ES6
- Drop string split and string substring from proposal
- Define API
- Eric: provides use case of counting words.
- Example: “   hello world” – two segments or four? Some use cases require four segments; word counting only two.

2.7 Timezone Offsets

http://wiki.ecmascript.org/doku.php?id=globalization:timezoneoffsets

Use Case: Recurring calendar event, needs to handle daylight savings shift, possibly across multiple time zones (schedule in US Pacific Time, attendee in central Europe...).

- getTimeZoneOffset does not handle all time zones today
- use calendar, time zone, day
- Adding arguments to existing methods creates problem for feature detection, although toLocaleString has shown that there are ways (new versions throw exceptions for invalid argument values).

All participants were okay with discussing this time zone offsets proposal for possible inclusion in second edition.

Action item: Flesh out time zone offsets proposal (Jungshik, Nebojša)

2.8 Number Parsing

http://wiki.ecmascript.org/doku.php?id=globalization:numberparsing

- Degree of leniency in number parsing: Should input in non-default number system be supported? Incorrect grouping?
- Error handling: Should non-number characters be ignored or treated as error?

Decisions:

- Expect only numbers
- If there are grouping separators they have to be in proper positions or completely missing

Action item: Update the proposal (Nebojša).

2.9 DisplayName

http://wiki.ecmascript.org/doku.php?id=globalization:displayname

API to obtain display names for locale and its components such as languages, regions and scripts

- Get back full locale name, language name, region name, script name
- Fallback is English name
- short/colloquial name
- long/official name (bad in some cases, e.g. Taiwan)

Action items:

- Comments on wiki (All)
- Use cases to be documented (Jungshik)