ECMAScript Internationalization API

Norbert Lindenberg
Internationalization
Ad-hoc Group

• Google: Nebojša Ćirić, Mark Davis, Mark Miller, Roozbeh Pournader, Markus Scherer, Jungshik Shin, John Tamplin

• IBM: Steven Loomis

• Microsoft: Eric Albright, Peter Constable, Shawn Steele

• Mozilla: Zbigniew Braniecki, Allen Wirfs-Brock

• Invited experts: Richard Gillam, Norbert Lindenberg, Addison Phillips, Nicholas Zakas
API Overview
Goals

• Provide most commonly used internationalization functionality, with configuration options
• Support multiple locales per application
• Leverage existing implementations
• Results matching user expectations
• Based on standard identifiers
• Compatible with ES 5.1 and 6
Functionality

- Locale negotiation
- Collation (string comparison)
- Number formatting
- Date and time formatting
Main Usage Patterns

• coll = new Collator(localeList, options);
  a.sort(coll.compare);

• format = new Format(localeList, options);
  result = format.format(valueToFormat);

• result = date.toLocaleString(localeList, options);
Locale and Options Negotiation

- Complicated by Unicode extension
  - de-u-co-phonebk-cu-jpy-nu-thai
  - Keys are not arranged generic->specific, so no fallback by chopping subtags
- Some parameters are locale related
- Some parameters should/must be under application control (currency)
- One parameter (co) has both
Locale and Options Negotiation

- Constructors combine locale and options negotiation
- ResolveLocale splits Unicode extension from core language tag
- Two algorithms for core:
  - BCP 47 Lookup (fully specified)
  - Best-fit (implementation dependent)
- Locale data based negotiation for extension keys and options
resolvedOptions

- resolvedOptions accessor
- actual locale
- actual parameter settings
Implementation

Dependencies

- Modeling the real world, with different results
  - Set of supported locales
  - Set of supported features per locale
  - Collation rules
  - Number formats, currency symbols
  - Calendar and time zone rules
  - Date and time formats
Implementation

Dependencies

- Different capabilities of implementations
  - Best-fit language negotiation
  - Best-fit date-time format negotiation
  - Collation features
    - Combinability of locales and calendars
    - Time zone selection
Signature ①

- Intl
- Intl.LocaleList([locales])
- Intl.LocaleList.length & indexed properties
Signature

- Intl.Collator ([localeList [, options]])
- Intl.Collator.supportedLocalesOf (requestedLocales [, options])
- Intl.Collator.prototype.compare(x, y)
- Intl.Collator.prototype.resolvedOptions
Signature

- `Intl.NumberFormat([[localeList [, options]]])`
- `Intl.NumberFormat.supportedLocalesOf(requestedLocales [, options])`
- `Intl.NumberFormat.prototype.format(x)`
- `Intl.NumberFormat.prototype.resolvedOptions`
- *Same for Intl.DateTimeFormat*

*Same for Intl.DateTimeFormat*
Enhanced Core ECMAScript Functions

- Respecify *locale* methods
  - String.prototype.localeCompare
  - Number.prototype.toLocaleString
  - Date.prototype.toLocaleString & Co.
- Add localeList, options parameters to all
Specification Update

• Minimum Unicode 5.0
• Check for “structurally valid” extensions
• Removed kb, kc, kh options from Collator
• Fixed ToDateTimeOptions: allow time only
• Use Record specification type
Pending Issues

- Numbering system – waiting for CLDR spec update
- Collator sensitivity – naming, descriptions
- List of default locales
- Support for likely subtags
- Options in resolved locale tag
- Bound format methods
- Fall-back locale data normative
Implementations

- Microsoft
- Google
- others?
- Issue: prefix such as V8Intl, MSIntl, or hope that API doesn’t change anymore? Opt-in for ES-Lang function update?
Test Suite

- Google has implemented initial test suite
- Needs more in-depth testing
  - locale negotiation
  - string comparison
  - number formatting
  - date and time formatting
Community Feedback

- Don’t like to pass around locale list
- No good place to hang locale for embedded context
- Security issue
- Would like pattern strings for date format
- Postponed to future version
Time Line

- End of March: Review of complete API, and TC39 approval of feature complete "implementation draft".
- April-June: Trial implementations and development of test suite.
- April-May: Detailed technical review and polishing of the specification, including feedback from TC39 members, testers, and implementers.
- May 1: Public review draft available for TC39
- May: TC39 meeting approves draft for public review, announces availability of draft, and requests feedback from other web stakeholders.
- July: TC39 meeting reviews feedback, approves any changes.
- September: TC39 meeting approves final draft for submission to the GA.
- December: Ecma GA approval.