

Implement “Intl Locale Info” Proposal to V8 Design Doc

[public viewable design doc]

Status: Draft

Author: ftang@chromium.org

Last Updated: 2021-12-03

Objective

Implement the ECMA402 “[Intl Locale Info](#)” proposal in [v8 JavaScript engine](#).

Background

“[Intl Locale Info](#)” (shortened as “target proposal” in this document) is a new proposal, promoted to Stage 3 during the [TC39 2021-04 meeting](#), part of the [ECMA-402 ECMAScript Internationalization API](#) program. ICU4C library already implements related functionality in C++ and this project implements the necessary adopting code in V8 to bind the JavaScript API to the C++ implementation.

Overview

The “[Intl Locale Info](#)” adds 7 new get only methods to the pre-existing Intl.Locale object.

Detailed Design

Flag

Since the target proposal is still under Stage 3, we guard the code under flag. All observable behavior is controlled by `--harmony_intl_locale_info`. This flag will be added to [src/flags/flag-definitions.h](#) under **HARMONY_INPROGRESS**.

BuiltIns Code

We need to add the following declarations to [src/builtins/builtins-definitions.h](#)

- LocalePrototypeCalendars
- LocalePrototypeCollations

- LocalePrototypeHourCycles
- LocalePrototypeNumberingSystems
- LocalePrototypeTextInfo
- LocalePrototypeTimeZones
- LocalePrototypeWeekInfo

Also, we need to add the implementations to <src/v8/src/builtins/builtins-intl.cc>

These implementations will forward the code to JSLocale

- LocalePrototypeCalendars will call JSLocale::Calendars()
- LocalePrototypeCollations will call JSLocale::Collations()
- LocalePrototypeHourCycles will call JSLocale::HourCycles()
- LocalePrototypeNumberingSystems will call JSLocale::NumberingSystems()
- LocalePrototypeTextInfo will call JSLocale::TextInfo()
- LocalePrototypeTimeZones will call JSLocale::TimeZones()
- LocalePrototypeWeekInfo will call JSLocale::WeekInfo()

Bootstrapping Code

We need to change <src/init/bootstrapper.cc> to add the implementation of a new function `Genesis::InitializeGlobal_harmony_intl_locale_info()` to connect the interface to the BuiltIns code above.

Additional Symbols

To implement the target proposal, we also need to add the following symbols to <src/init/heap-symbols.h>:

- “calendars”,
- “collations”,
- “direction”,
- “firstDay”,
- “hourCycles”,
- “minimalDays”,
- “numberingSystems”,
- “textInfo”,
- “timeZones”
- “weekend”,
- “weekInfo”,

Add new methods to JSLocale class

The pre-existing class JSLocale will be implemented in the following new methods:

- JSLocale::Calendars()
- JSLocale::Collations()

- JSLocale::HourCycles()
- JSLocale::NumberingSystems()
- JSLocale::TextInfo()
- JSLocale::TimeZones()
- JSLocale::WeekInfo()

JSLocale::Calendars() & JSLocale::Collations()

The implementation of Calendars() and Collations() depends on and delegate the call to [icu::Calendar::getKeywordValuesForLocale\(\)](#) and [icu::Collator::getKeywordValuesForLocale\(\)](#) to find out the list of calendars and collations for each locale. The operation and are very similar, therefore we abstract the implementation into a template

```
template <typename T>
MaybeHandle<JSArray> GetKeywordValuesFromLocale(
    Isolate* isolate, const char* key, const char* unicode_key,
    const icu::Locale& locale,
    const std::map<std::string, std::string>& filters)
```

Notice for calendar, we need to replace the return value of "gregorian" to "gregory", and "ethiopic-amete-alem" to "ethioaa". For collations, we need to filter out "standard" and "search".

JSLocale::HourCycles()

The implementation of HourCycles() depends on and delegates the call to [icu::DateTimePatternGenerator::getDefault\(\)](#) .

JSLocale::NumberingSystems()

The implementation of NumberingSystems() depends on and delegates the call to pre-existing [Intl::GetNumberingSystem\(\)](#)

JSLocale::TextInfo()

The implementation of TextInfo() depends on and delegates the call to [uloc_getCharacterOrientation\(\)](#).

JSLocale::TimeZones()

The implementation of TimeZones() depends on and delegates the call to [icu::TimeZone::createTimeZoneIDEnumeration\(\)](#) .

JSLocale::WeekInfo()

The implementation of TimeZones() depends on and delegates the call to [icu::Calendar::getFirstDayOfWeek\(\)](#), [icu::Calendar::getDayOfWeekType\(\)](#), and [icu::Calendar::getMinimalDaysInFirstWeek\(\)](#).

Testing Code

See the “[Testing Plan](#)” section for more detail.

- [v8/test/intl/locale](#)

```
locale/locale-info-timezones-sorted.js
locale/locale-collations.js
locale/locale-info-check-property.js
locale/locale-info-ext.js
locale/locale-numberingSystems.js
locale/locale-info-check-return-types.js
locale/locale-info-no-undefined.js
locale/locale-calendars.js
```

- test262- feature in test262 - Intl.Locale-info
 - Tests: found by “\$ find test/test262/data/test/intl402/|xargs egrep -s Intl.Locale-info|cut -d":" -f1|cut -c24-1000”

```
intl402/Locale/prototype/numberingSystems/prop-desc.js
intl402/Locale/prototype/numberingSystems/name.js
intl402/Locale/prototype/numberingSystems/output-array.js
intl402/Locale/prototype/numberingSystems/branding.js
intl402/Locale/prototype/collations/prop-desc.js
intl402/Locale/prototype/collations/name.js
intl402/Locale/prototype/collations/output-array.js
intl402/Locale/prototype/collations/branding.js
intl402/Locale/prototype/collations/output-array-values.js
intl402/Locale/prototype/calendars/prop-desc.js
intl402/Locale/prototype/calendars/name.js
intl402/Locale/prototype/calendars/output-array.js
intl402/Locale/prototype/calendars/branding.js
intl402/Locale/prototype/hourCycles/prop-desc.js
intl402/Locale/prototype/hourCycles/name.js
intl402/Locale/prototype/hourCycles/output-array.js
intl402/Locale/prototype/hourCycles/branding.js
intl402/Locale/prototype/hourCycles/output-array-values.js
```

```
intl402/Locale/prototype/weekInfo/prop-desc.js
intl402/Locale/prototype/weekInfo/name.js
intl402/Locale/prototype/weekInfo/branding.js
intl402/Locale/prototype/weekInfo/output-object.js
intl402/Locale/prototype/weekInfo/output-object-keys.js
intl402/Locale/prototype/textInfo/prop-desc.js
intl402/Locale/prototype/textInfo/name.js
intl402/Locale/prototype/textInfo/branding.js
intl402/Locale/prototype/textInfo/output-object.js
intl402/Locale/prototype/textInfo/output-object-keys.js
intl402/Locale/prototype/timeZones/prop-desc.js
intl402/Locale/prototype/timeZones/name.js
intl402/Locale/prototype/timeZones/output-array.js
intl402/Locale/prototype/timeZones/branding.js
intl402/Locale/prototype/timeZones/output-undefined.js
intl402/Locale/prototype/timeZones/output-array-sorted.js
```

External Dependency

Require Changes In ICU:

- None

Completion of test262:

- DONE

Open/Close Issues (OI/CI)

Open Issues (OI)

- None

Closed Issues (CI)

- Weekend expression for place with non continuous weekend
 - <https://github.com/tc39/proposal-intl-locale-info/pull/44>
 - <https://chromium-review.googlesource.com/c/v8/v8/+3307685>

Project Information

- **Design Doc (this doc):** [this document](#)
- **Tracking Bug:** [v8:11638](#)
- **Chrome Platform Status dashboard:** [feature/5566859262820352](#)
- **Specification Champion:** Frank Tang
- **TL:** Frank Tang ftang@google.com
- **TLM:** Nebojša Ćirić cira@google.com
- **SWE/Developer:** Frank Tang ftang@google.com
- **Reviewers:**
 - Jakob Kummerow jkummerow@chromium.org
 - Shu-yu Guo syg@chromium.org
- **CL:**
 - Most implementation code: [v8/+2570218](#) **[DONE]**
 - Flip to Stage in m92: [v8/+2878908](#) **[DONE]**
 - Weekend changes - [v8/+3307685](#) **[DONE]**
 - Ship for m99 - [v8/v8/+3315223](#) **[In Progress]**
- **Discussion Hangout:** [Intl-v8 group hangout](#)
- **Ready to Trial**(May 7, 2021)
<https://groups.google.com/a/chromium.org/g/blink-dev/c/5spmAncbooE/m/NdwZGjLpAgAJ> **[DONE]**
- **Intent to Ship:** [blink-dev/c/4ZMt5ukQNRs/m/0fIHnuaBBgAJ](#) **[DONE]**
- **LGTM:** chrisht@chromium.org, miketaylr@chromium.org, tkent@chromium.org

Testing Plan

How to Run JavaScript Unit Tests

```
$ tools/dev/gm.py x64.release
$ tools/run-tests.py --outdir=out/x64.release \
  intl/locale/locale-info* test262/intl402/Locale/prototype*
```

Work Estimates

Schedule

- **Code complete:** May 2021 **[DONE]**
- **Stage:** Mid May 2021 - m92 [2878908](#) **[DONE]**
- **Ship: Target for m99**
 - Flip bit in mid Dec 2021 **[In progress]**
 - Stable release in March 2022

Reference Documents & Site

- About the Specification
 - <https://github.com/tc39/proposal-intl-locale-info>
 - <https://tc39.es/proposal-intl-locale-info/>
- TC39 update about the proposal
 - Advanced to [Stage 1](#) in TC39 2020-09 meeting.
 - Advanced to [Stage 2](#) in [TC39 2021-01-25~28 meeting](#).
 - Jan 2021 TC39 meeting approve to move to Stage 2 WITH the condition to drop unitInfo
 - 2021-04-08 ECMA402 meeting, we agreed to propose to TC39 for Stage 3.
 - Advanced to [advance to Stage 3](#) in [TC39 2021-04 meeting](#).
 - [Update during Stage 3](#) in [TC39 2021-07 meeting](#).
 - [Update during Stage 3](#) in [TC39 2021-10 meeting](#)
 - [Update during Stage 3](#) in TC39 2021-12 meeting
- About V8
 - [V8 Official Doc](#)
 - [Checkout V8 Source](#)
 - [Using Git to get V8](#)
 - [Building w/ GN](#)
 - [Testing in V8](#)
 - [V8 Embedder Guide](#)
 - [V8 Development How To](#) (Additional Info)
 - [Search Code in V8](#)
- About ICU4C
 - ...
- About ECMAScript
 - [ECMA-262 ECMAScript2019 Specification](#)
 - [ECMA-402 ECMAScript 2018 Internationalization API Specification](#)
 - Test suite for ECMA-262 <https://github.com/tc39/test262>
- About git
 - [git Tutorial](#)