INTERNATIONALI[SZ]ATION FOR LOCALIZATION (i18n for l10n)

Eike Rathke

Software Engineer

Sun Microsystems

Internationalization Myths

> “My product uses open source and so internationalization requirements don't apply.”

Myth #5, The I18n G.A.L.
http://blogs.sun.com/roller/page/i18ngal?entry=myth_5_for_open_source

Agenda

Terms

History

- class International, Numberformatter, i18n framework

Standards

- ISO 639, ISO 15924, ISO 3166, RFC 3066

Today

- Services, ICU, shortcomings

ToDo

- Next, near future, medium future, far future

Glossary - Terms

- locale [lO-'kal]
  - Combination of language plus region/country/culture
- globalization (g11n)
  - The overall process
- internationalization (i18n)
  - Abstract out local details
    - Prepare software such that it runs independent of locale assumptions with different locales
- localization (l10n)
  - Specify details for a particular locale

Ideal Internationalized Program

- Same executable can run worldwide
- No hardcoded UI messages or labels
- Culturally-dependent data localized
- Support for new languages does not require recompilation
  - OOo: no recompilation, but build resources in tree
- Can be localized quickly
  - OOo: does take its time

Culturally Dependent Data

Messages

Labels on GUI components

Online help

Sounds

Colors

Graphics

Icons

Dates

Times

Numbers

Currencies

Measurements

Phone numbers

Honorifics and personal titles

Postal addresses

Page layouts

History - class International

- tools/inc/intn.hxx
  - tools/source/intntl/intn{,2,lang,tab}.cxx
    - Table data hard-coded into the source code
      - LanguageTable: day and month names of Gregorian calendar, quotation marks, pointers to character handling specific functions like upper/lower case, compare; language centric
      - FormatTable: separators and all information needed for number formatting; country centric
        - Only on Windows®: merged-in system data from Regional Settings
      - pros: flexible because every single data item was exchangeable during runtime
      - cons: hard to maintain, full functionality on Windows® only, LCID centric

Microsoft® Locale Identifier (LCID)

- 16-bit value
  - Lower 10 bits primary language ID
  - Upper 6 bits sub-language ID
  - e.g. primary 0x09 combined with secondary 0x01 == (0x01 << 10) | 0x09 == 0x0400 | 0x09 == 0x0409
  - User-definable value ranges
    - primary: 0x0200 to 0x03FF
    - secondary: 0x20 to 0x3F
    - all other values reserved for Windows® system use
    - e.g. (0x01 << 10) | 0x022B == 0x062B
  - More details in comment of tools/inc/lang.hxx

Numberformatter Legacy

- Predefined format codes
  - Fixed meaning of format indices
    - NUMBER_INT (index 1), NUMBER_DEC2 (index 2)
  - Windows® Regional Settings followed in some formats
    - NUMBER_SYSTEM (index 5)
      DATE_SYSTEM_SHORT (index 18)
  - Settings obtained for separators and YMD order
    - DATE_SYS_DDMMYY (index 20)
      DATE_SYS_DDMMYYYY (index 21)
      - DATE_SYS_DDMMYY could be DD.MM.YY, MM/DD/YY, YY-MM-DD
        DATE_SYS_DDMMYYYY similar but with 4 digits year

constant's names in offapi/com/sun/star/i18n/NumberFormatIndex.idl
values in generated solver inc/com/sun/star/i18n/NumberFormatIndex.hdl

History - Transition

- Transition to i18n framework
  - Focused on easy adoption by the applications
    - Similar data layout
    - Almost identical method names and functionality provided by intermediate layer, unotools/inc/*wrapper.hxx unotools/source/i18n/*.cxx
  - Parallel worlds of OpenOffice.org / StarOffice
    - Module i18n: basic implementation for OOo, more sophisticated implementation for SO based on proprietary code and data
    - Successive implementation of CJK functionality in module i18npool, emptying proprietary module i18n

Glossary - Standards

ISO 639 language codes

- ISO 639-1 Alpha-2 code
- ISO 639-2 Alpha-3 code
  - ISO 639-2/B for bibliographic use
  - ISO 639-2/T for terminological use, used in OOo
- ISO 639-3 Alpha-3 code for comprehensive coverage of languages (end of 2006)
- ISO 639-4 Implementation guidelines and general principles for language coding (planned, 2007?)
- ISO 639-5 Alpha-3 code for language families and groups (planned, 2008?)

Glossary - Standards

- ISO 15924 script codes, Alpha-4 and Numeric-3
  - e.g. Latn / 215, Cyrl / 220; not yet supported by OOo
- ISO 3166 country codes
  - ISO 3166-1 Alpha-2, public part, used by OOo
    - e.g. SI, DE, ZA
  - ISO 3166-1 Alpha-2, Alpha-3, Numeric-3, commercial
    - e.g. ZA, ZAF, 710, South Africa, Republic of South Africa
  - ISO 3166-2 subdivision (region) codes
    - e.g. SI-01, DE-HH, ZA-WC

Glossary - Standards

ISO 4217   currency codes, Alpha-3 and Numeric-3

- e.g. EUR / 978, USD / 840; OOo uses Alpha-3

ISO 8601   date and time representation

- e.g. 2005-09-29T10:45

Unicode   character coding system

- Unique number for every character
  - no matter what the platform
  - no matter what the program
  - no matter what the language
    - http://www.unicode.org/standard/WhatIsUnicode.html

Glossary - Standards - RFC 3066

RFC 3066 tags for the identification of languages

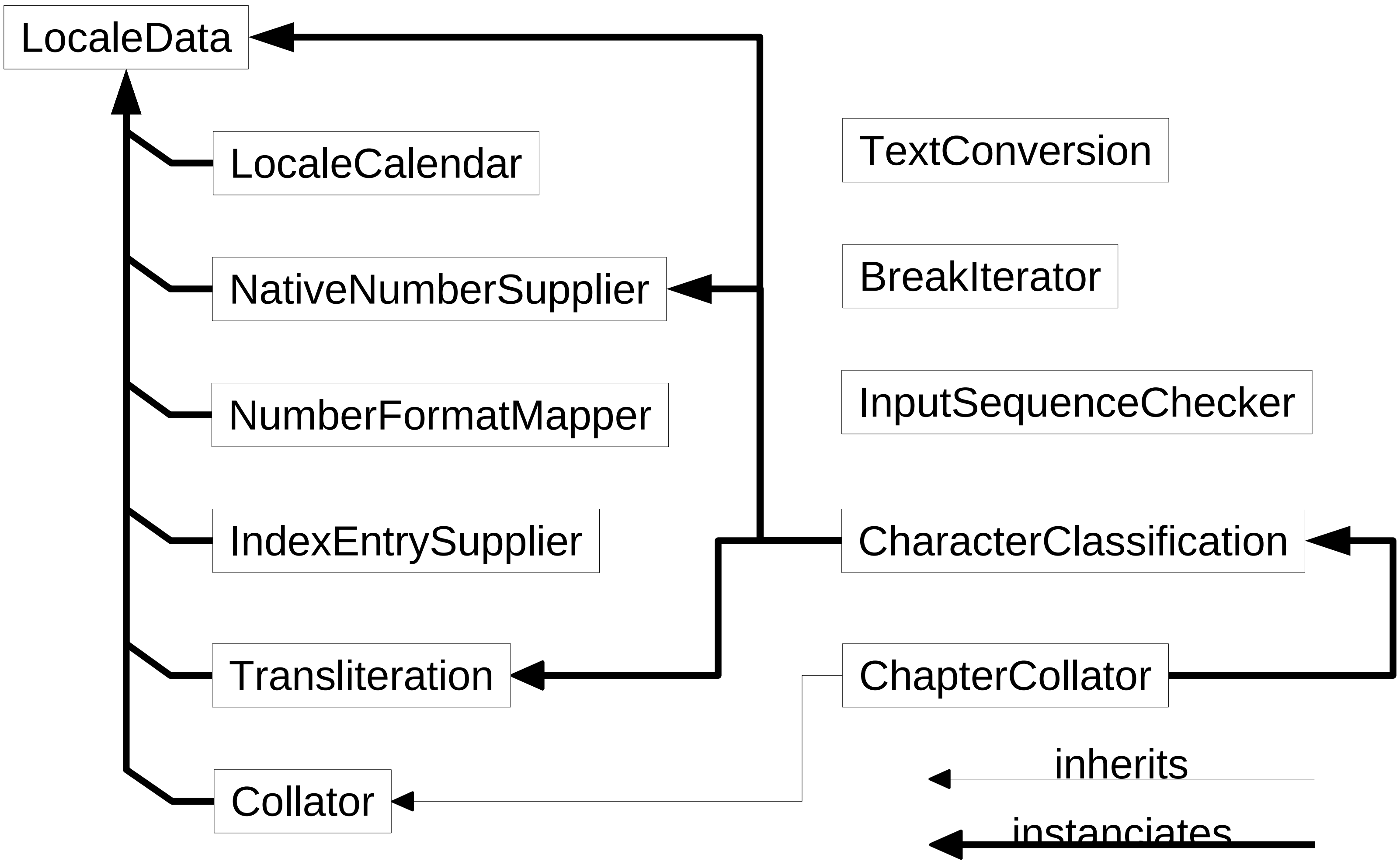
- primary-subtag
  - ISO 639-1
  - ISO 639-2
  - "i-something" IANA registered language; not supported by OOo
  - "x-something" private use; not supported by OOo
- second subtag
  - ISO 3166 alpha-2
  - 3 to 8 letters IANA registered
    - e.g. primary-second: sl-nedis, sl-rozaj, sr-Cyrl, sr-Latn; not in OOo
- subsequent subtags, country/region/dialect/variant

Glossary - Standards - RFC 3066bis

RFC 3066bis planned successor of RFC 3066

- More detailed view later

Today - Services Overview



What OOo Uses From ICU

Unicode data, character types, script types

Breakiterator

Rule based collator

Glyph layout engine

Calendar

Not used:

- locale data, encoding conversions, string functionality, number formatting

Shortcomings of Framework

- Design legacy
  - started as a replacement of class International
  - to support the existing code of the applications
- Published API not easily extensible, old API has to be kept stable and maintained
  - new methods only via optional interfaces
  - struct LocaleDataItem can't change size
  - enum UnicodeScript without "supersizer" can't be extended

ToDo - Next

- Alignment with CLDR (Common Locale Data Repository)
  - LocaleDataAudit_OOo_CLDR.html
  - Align OOo to CLDR
    - with help of tools that merge-in CLDR data
    - first set of ~15 locales in OOo2.0
    - most remaining locales for OOo2.0.1
  - Align CLDR to OOo
    - needs filing bugs against CLDR and providing "evidence"

ToDo - Near Future

Upgrade to ICU 3.4 / 3.6

- Will eliminate almost all patches currently applied to 2.6
  - goal of using system's ICU is nearer
- Better support of glyph layout for Indic languages
- Upstream ICU 3.6 will incorporate OOo patches for Khmer and Tibetan / Dzongkha
- Some minor annoyances removed
  - sr_YU kludge instead of sr_CS not necessary anymore
  - sh_YU kludge could become sr_Latn_CS if OOo supported sr_Latn as language with script identifier

ToDo - Medium Future

RFC 3066bis and draft ietf-ltru

- Successor of RFC3066
- Internet Engineering Task Force Language Tag Registry Update
- http://www.inter-locale.com/ID/why-rfc3066bis.html
- http://www.ietf.org/html.charters/ltru-charter.html
- language_country => language_[script]_region initially conforming to ISO 639, ISO 15924, ISO 3166
- Stability and accessibility of the underlying ISO standards not guaranteed => registration with IANA
  - e.g. ISO 3166 code CS was reused by ISO

ToDo - Future

- Separate string resources from build process
- Genitive month names in date formats
  - CLDR already has the data, OOo needs to adopt it
  - LocaleCalendar XCalendar::getDisplayName() must support it
  - Numberformatter must support it
  - Other code places maybe as well
- Support for plural forms

URLs

There's only one you really need to bookmark:

http://www.erack.de/bookmarks/D.html#i18n
has it all and will be continuously updated.

INTERNATIONALI[SZ]ATION FOR LOCALIZATION (i18n for l10n)

Eike Rathke

erack@sun.com