ICU Update

Yoshito Umaoka (IBM)
Markus W. Scherer (Google)

Agenda

- Isn't Unicode enough?
- Why ICU?
- Where is ICU?
- What's new in ICU 4.4?
- What's next for ICU?

Isn't Unicode enough?

- The nature of Unicode
- Internationalization, Localization & Locales

The nature of Unicode

- Handles all modern world languages
- Efficient and effective processing
- Lossless data exchange
- Enables single-binary global software

But…

- 1,400 pages + Annexes + additional standards
- More than 100,000 characters
- Major update every 3 years, minor update about once a year
- 80+ character properties, many multi-valued
- Affects many process: display, line-break, regular expressions…

Internationalization, Localization & Locales

Requirements vary widely across languages & countries

- Sorting
- Text searching
- Bidirectional text processing and complex text layout
- Date/time/number/currency formatting
- Codepage conversion
- … and so on

Performance is key

- It might be easy to do the right thing
- It is hard to do it fast

Why ICU?

- ICU Features
- ICU Works Everywhere

ICU Features

- Unicode text handling
- Charset conversions (175+)
- Charset detection
- Collation & Searching
- Locales from CLDR (450+)
- Resource Bundles
- Calendar & Time zones
- Complex-text layout engine
- Unicode Regular Expressions
- Breaks: word, line, …
- Formatting
  - Date & time
  - Durations
  - Messages
  - Numbers & currencies
  - Plurals
- Transforms
  - Normalization
  - Casing
  - Transliterations

ICU Works Everywhere

Mature, widely used set of C/C++ and Java libraries

- Basis for Java 1.1 internationalization, but goes far beyond Java 1.1

Very portable – identical results on all platforms/programming languages

- C/C++ (ICU4C): 30+ platforms/compilers
- Java (ICU4J): Oracle and IBM JRE

Full threading model

Customizable & Modular

Open source (since 1999) – but non-restrictive

- Governed by a Project Management Committee
- Contributions from many parties (IBM, Google, Apple, Yahoo, ...)

Where is ICU?

- ICU in IBM and Google
- Other ICU Users

ICU in IBM

- All 5 major software brands
- IBM operating systems
- Products

  Ascential Software, Cognos, PSD Print Architecture, DB2, COBOL, Host Access Client, InfoPrint Manager, Informix GLS, iSeries, Language Analysis Systems, Lotus Notes, Lotus Extended Search, Lotus Workplace, WebSphere Message Broker, NUMA-Q, OTI, OmniFind, Pervasive Computing WECMS, Rational Business Developer and Rational Application Developer, SS&S Websphere Banking Solutions, Tivoli Presentation Services, Tivoli Identity Manager, WBI Adapter/ Connect/Modeler and Monitor/ Solution Technology Development/WBI-Financial TePI, Websphere Application Server/ Studio Workload Simulator/Transcoding Publisher, XML Parser.

ICU in Google

- Web Search
- Chrome
- Android
- Adwords
- Google Finance
- Google Maps
- Blogger
- Google Analytics
- Google Gears
- Google Groups
- others...

Other ICU Users

ABAS Software, Adobe, Amazon (Kindle), Amdocs, Apache (Harmony, Lucene, Solr, PDFBox, Tika, Xlan, Xerces, ...), Appian (Mac OS X, iPhone, Safari, iTunes for Windows), Apple, Argonne National Laboratory, Avaya, BAE Systems Geospatial eXploitation Products, BEA, BluePhoenix Solutions, BMC Software, Boost, BroadJump, Business Objects, caris, CERN, Debian Linux, Dell, Eclipse, eBay, EMC Corporation, ESRI, Free BSD, Gentoo Linux, GroundWork Open Source, GTK+, Harman/Becker Automotive Systems GmbH, HP, Hyperion, IBM, Inktomi, Innodata Isogen, Informatica, Intel, Interlogics, IONA, IXOS, Jikes, Library of Congress, Mathworks, Mozilla, Netezza, OpenOffice, Lawson Software, Leica Geosystems GIS & Mapping LLC, Mandrake Linux, OCLC, Progress Software, Python, QNX, Rogue Wave, SAP, SIL, SPSS, Software AG, Sun Microsystems (Solaris, Java), SuSE, Sybase, Symantec, Teradata (NCR), Trend Micro, Virage, webMethods, Wine, WMS Gaming, XyEnterprise, Yahoo!, and many others.

What's new in ICU 4.4?

- Release Summary
- Normalizer2
- Resource Bundle Optimization
- Big Decimal Support (C)
- Select Format
- 64bit Time Zone Transitions
- Java 5 Syntax
- ICU4J source package structure

Release Summary

http://icu-project.org/download/4.4.html

- 2010-03-17 ICU 4.4 (C & J)
- 2010-04-28 ICU4.4.1 (C & J)
- 2010-06-23 ICU4J 4.4.1.1

ICU 4.4 Data

- Unicode 5.2
- Locale Data
  - CLDR 1.8 in ICU 4.4
  - CLDR 1.8.1 update in ICU 4.4.1
  - Over 22% more data than CLDR 1.7
  - Many new locales (African locales from Afrigen)

Normalizer2

- Unicode 5.2 adds 3rd form: NFKC_Casefold
- UTS #46/IDNA2008 combines mapping+normalization
- Other combinations useful
- ICU 4.4: New API allows custom data
- All operations with any data file
- Smaller files, select data for desired forms

Resource Bundle Optimization

- Smaller resource bundle files (-20%), smaller ICU data (-15%)
  - Reduced padding
  - Smaller table/array structures
  - Shared files for item keys
- Still memory-mapped
- No API change

ICU4C Big Decimal Support

- Decimal floating-point arithmetic used for financial & user-centric applications
- Java: BigDecimal;
  C/C++: http://speleotrove.com/decimal/
- ICU4C 4.4: Format & parse
- API: Decimal Floating Point strings

Select Format

- Selects message variants
  - Use case: Gender selection
  - “{gender,select,female{She walks.}other{He walks.}}”
- Similar to choice/plural
- One message string, one translator

64bit Time Zone Transitions

- Modified zic to generate time zone resource
- The time transition data was 32bit second (effective 1901-2038)
- For ICU 4.4 and beyond, ICU time zone resource includes the time transition data out of 32bit second range
- Changed data structure of other time zone resources at the same time

ICU 3.8.1 to 4.2.x	ICU 4.4 and later
zoneinfo.txt metazoneInfo.txt supplementalData.txt	zoneinfo64.txt metaZones.txt timezoneTypes.txt windowsZones.txt

Java 5 Syntax

- Changed the minimum Java Runtime requirement to Java 5
  - ICU4J up to 4.2 runs on JRE 1.4, can build a version supporting JRE 1.3 on JDK 1.3
  - 4.4 or later version do not support JRE 1.4 or older at all
- Adopted generics/co-variant type in ICU4J public API and implementation
- JDK parallel APIs have equivalent API signatures with JDK 1.6

ICU4J source package structure

- ICU4J up to 4.2 had single root for all classes
  - Spaghetti inter dependencies
  - Error prone modularization
- New source package structure
  - demos/ - demonstration programs
  - main/ - ICU4J runtime library and unit test
  - tools/ - ICU4J stand alone tools and build tools

ICU4J source package structure (cont.)

Structure under main/

```
classes/
        charset/        Implementation of Java charset service provider
        collate/        Collation service and its dependant
        core/           Unicode, formatting and other miscellaneous services
        currdata/       Currency localized name provider
        langdata/       Language localized name provider
        localespi/      Implementation of Java Locale service provider
        regiondata/     Region localized name provider
        translit/       Transliterator service and its dependant
shared/
        build/          Common component build scripts and properties
        data/           ICU data imported from ICU4C build
        license/        License files
tests/
        charset/        Test cases for charset service
        collate         Test cases for collation service
        core/           Test cases for Unicode, formatting and other services
        framework/      Common test framework and utilities
        localespi/      Test cases for locale SPI
        packaging/      Test cases for validating packaging without localized name providers
        translit/       Test cases for transliterator service
```

What's next for ICU?

- Release Schedule
- ICU 4.6 Release Summary

Release Schedule

- ICU 4.6
  - 2010-06-23 Milestone 1
  - 2010-09-29 Milestone 2
  - 2010-10-20 Release Candidate 1
  - 2010-11-10 Release
- ICU 4.8
  - 2011-05-04 Release

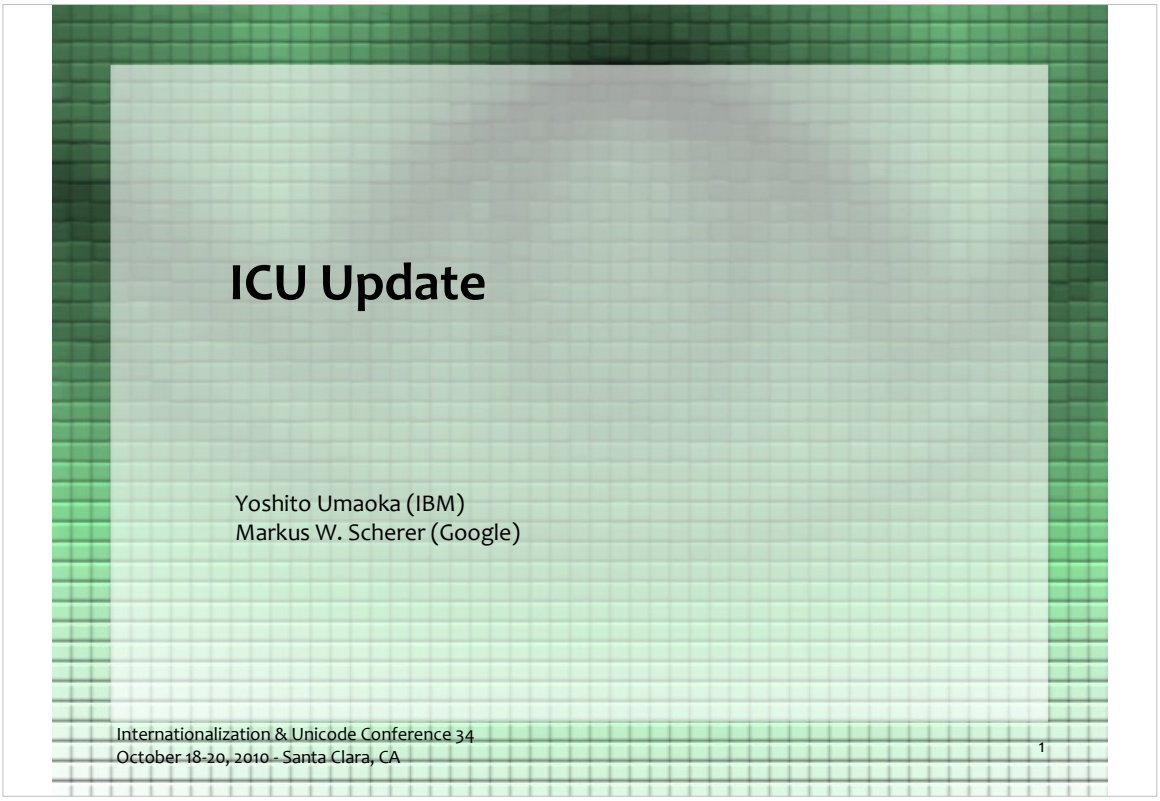
ICU 4.6 Release Summary

- Unicode 6.0
- CLDR 1.9
- Major Feature Items
  - UTS#46 Unicode IDNA compatibility processing support
  - Regular expression API enhancements (UText support, int64 index, find progress callback)
  - Alphabetic Index
  - BCP47 support update

References

ICU Main Site: http://icu-project.org

- Download ICU Releases
- User Guide
- Demonstrations
- Technical FAQ
- Bug Report
- Mailing Lists (design & support)



ICU (International Components for Unicode) is an open source development project sponsored, supported, and used by many organizations. It is dedicated to providing robust, full-featured, commercial quality, freely available Unicode-based technologies.

Comprehensive support for the Unicode Standard is the basis for multilingual, single-binary software. ICU uses the most current versions of the standard, and provides full support for supplementary characters.

As computing environments become more heterogeneous, software portability becomes more important. ICU lets you produce the same results across all the various platforms you support. It offers great flexibility to extend and customize the supplied system services.

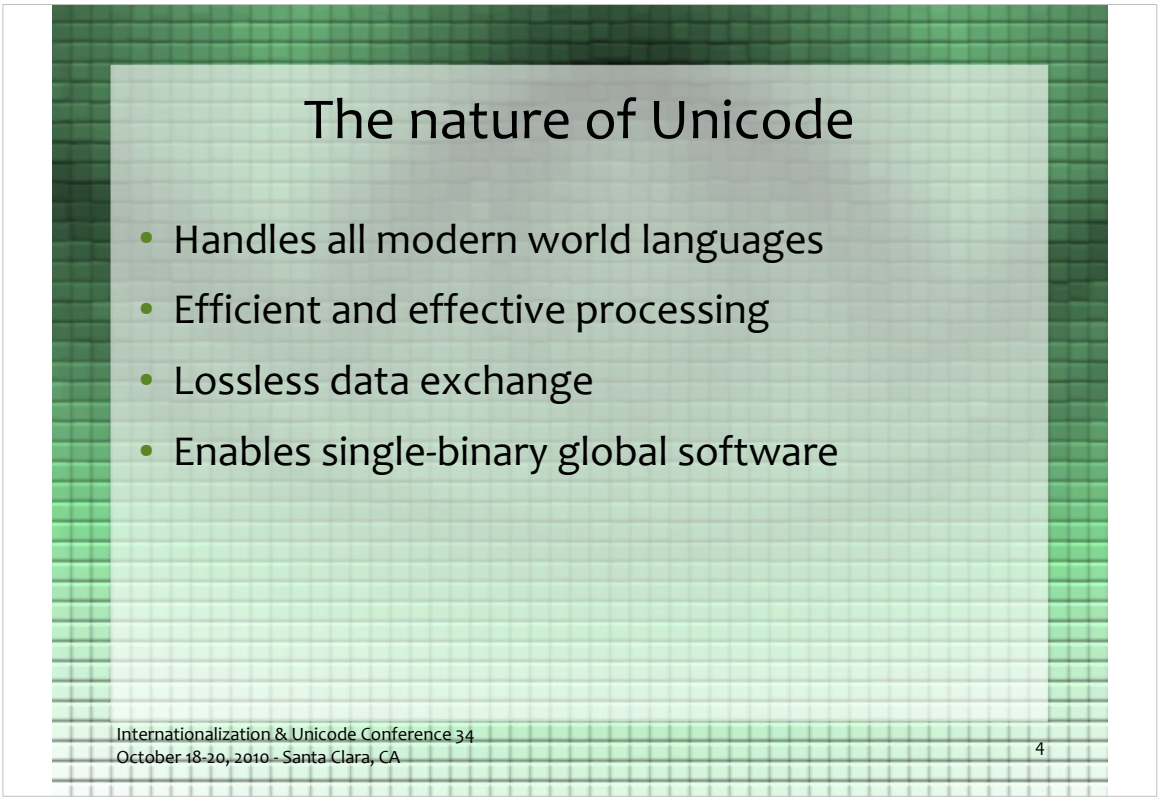
For more information, see the ICU website: http://icu-project.org

Agenda

- Isn't Unicode enough?
- Why ICU?
- Where is ICU?
- What's new in ICU 4.4?
- What's next for ICU?

Isn't Unicode enough?

- The nature of Unicode
- Internationalization, Localization & Locales



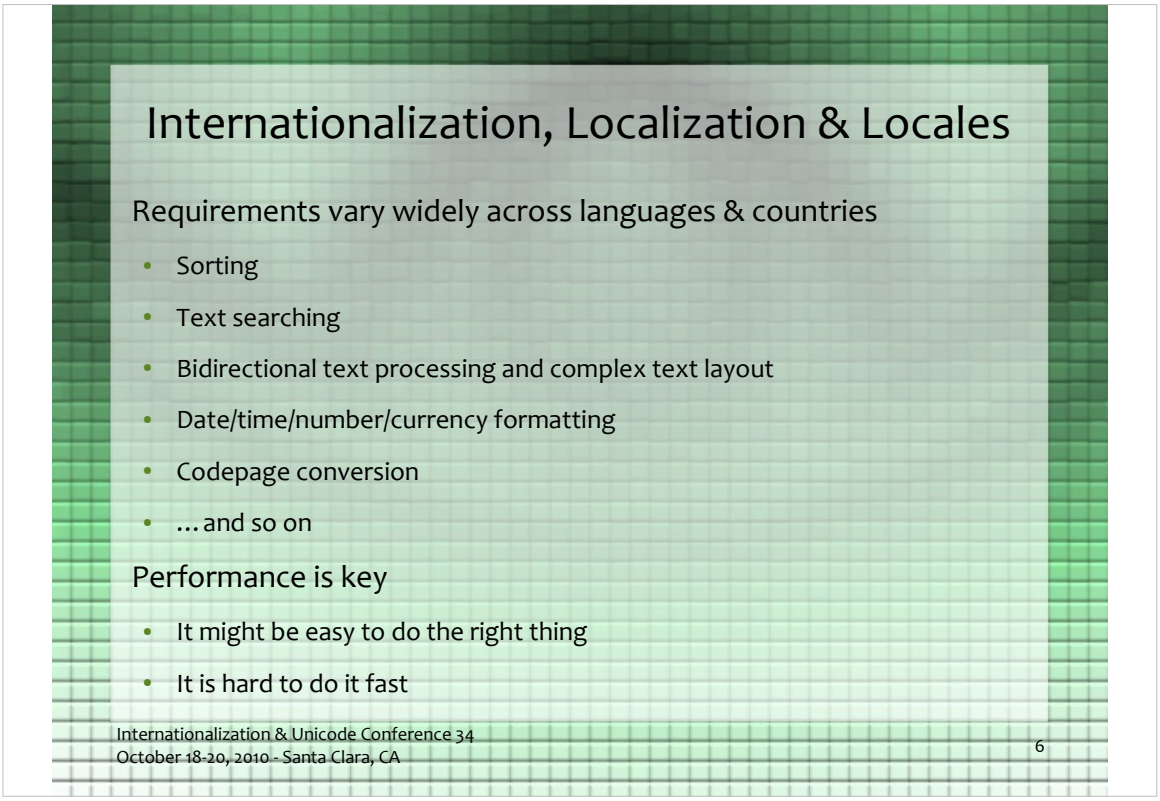
Unicode (and the parallel ISO 10646 standard) defines the character set necessary for efficiently processing text in any language and for maintaining text data integrity. In addition to global character coverage, the Unicode standard is unique among character set standards because it also defines data and algorithms for efficient and consistent text processing. This simplifies high-level processing and ensures that all conformant software produces the same results. The widespread adoption of Unicode over the last decade made text data truly portable and formed a cornerstone of the Internet.

Unicode enables lossless exchange of multilingual data between different types of computing systems, as well as single-binary installations of software which can handle text in all languages.

As a result, the Unicode Standard is complex and voluminous and not trivial to implement. ICU supports all Unicode characters, is regularly updated to the latest Unicode version, and implements and provides most of the properties and algorithms.

But…

- 1,400 pages + Annexes + additional standards
- More than 100,000 characters
- Major update every 3 years, minor update about once a year
- 80+ character properties, many multi-valued
- Affects many process: display, line-break, regular expressions…



The design and architecture of software that can work with multiple languages and cultural specializations is called internationalization. It involves taking into account a variety of attributes for many areas of text handling and data input and output. The most common attributes are the written language and the country or region for which data is processed or presented. Standard codes for these attributes, and sometimes others, are often combined into "locale identifiers". Depending on the context, the term "locale" refers either to such locale identifiers or to the relevant collection of associated data and behaviors.

In addition to these familiar attributes, others are also important and cannot be reliably inferred. For example, currency codes and codepages need to be identified reliably for correct results.

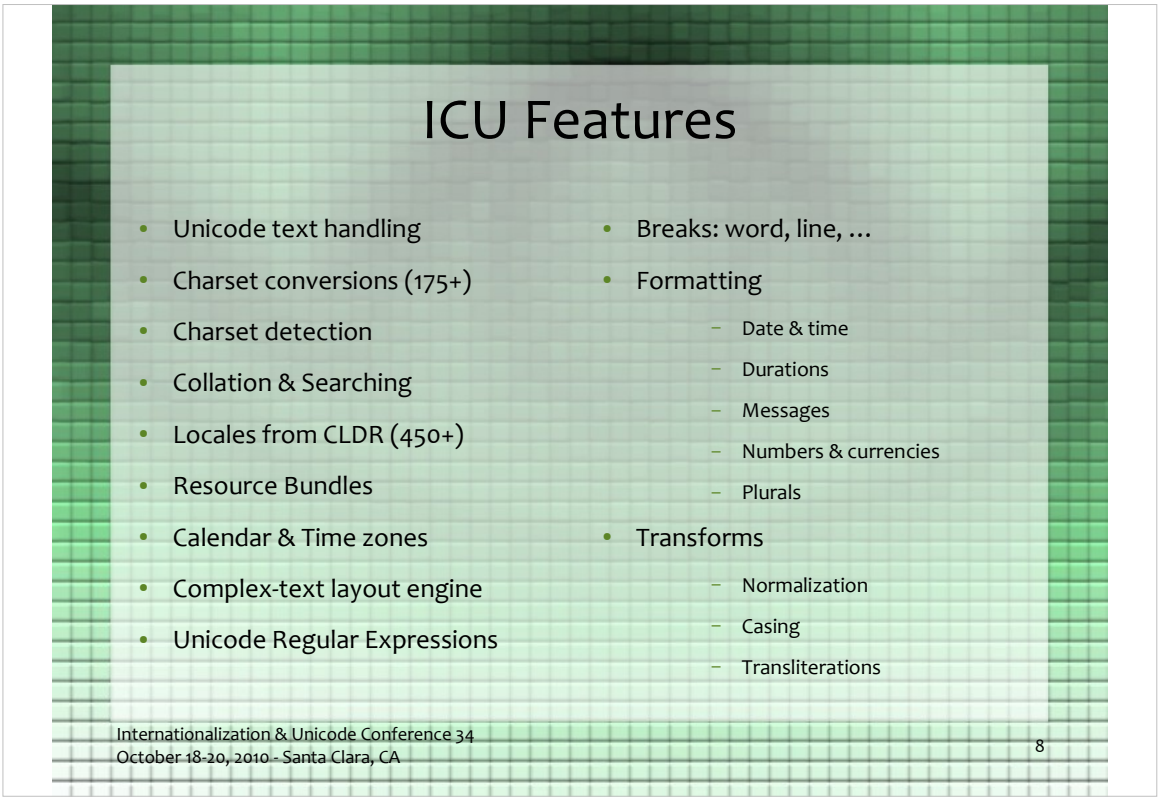
Localization provides internationalized software with locale-specific User Interface elements (text, images, layout) and sometimes functionality for regional business rules or similar.

The term globalization is sometimes used as a synonym for internationalization. We use it more narrowly, for software which can be compiled and installed once and handles text in all languages at the same time, as opposed to requiring recompilation for each locale. Such software needs to use Unicode for text processing.

It is often relatively easy to satisfy the requirements from one language or culture, or a small number of closely related ones. However, with the diversity of requirements from many languages and cultures on many processes, and the desire for high performance in many cases, the implementation of these processes can become rather complex. The ICU libraries provide "shrink-wrapped", reusable, tested implementations that were designed with performance in mind.

Why ICU?

- ICU Features
- ICU Works Everywhere

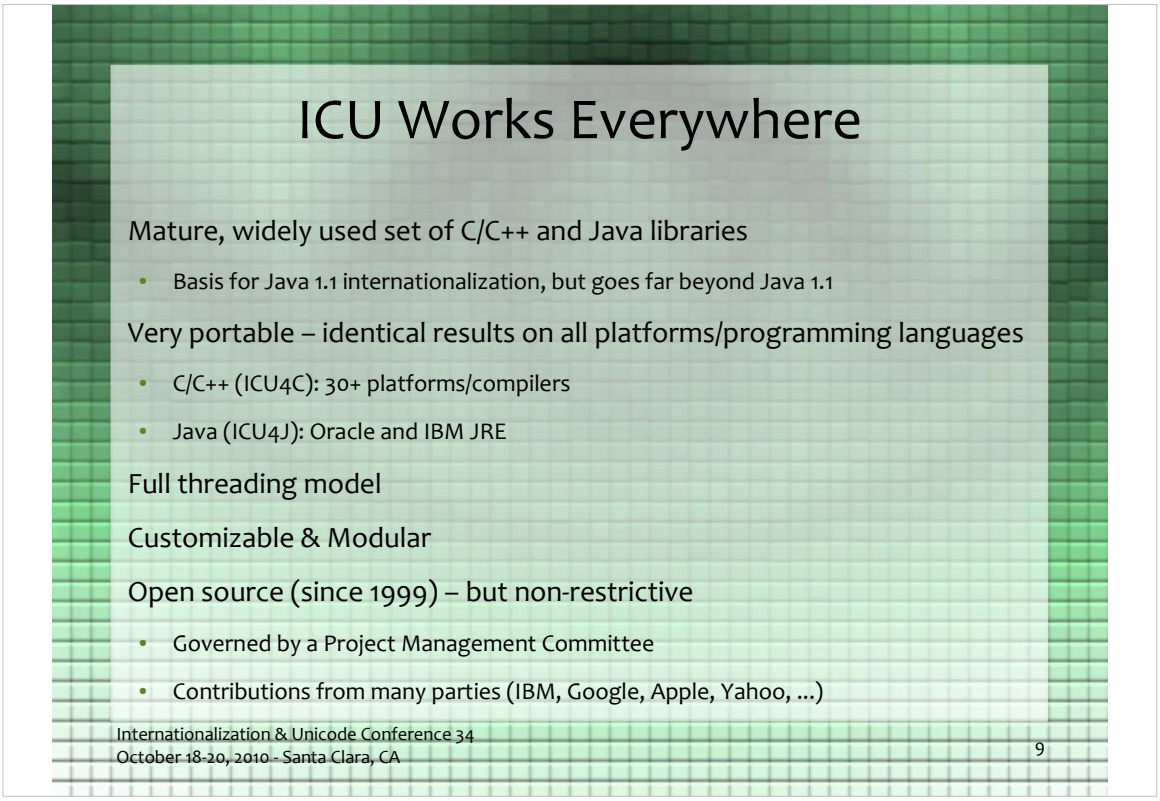


In addition to basic Unicode standard conformance, both the ICU Java library ("ICU4J") and the C/C++ libraries ("ICU4C") also provide a full set of internationalization features listed above.

Notes on C/C++ vs. Java

ICU C/C++ and Java APIs do differ slightly due to the differences of programming languages. Sometimes the feature development in ICU4C leapfrogs ICU4J or vice versa by 1-2 releases.

Since ICU is open source and closely tracks the Unicode Standard, ICU can support changes and additions to the Unicode Standard much more quickly than Java. Java support for Unicode is tied to major releases of the JDK, and can lag the Unicode Standard by a year or more.



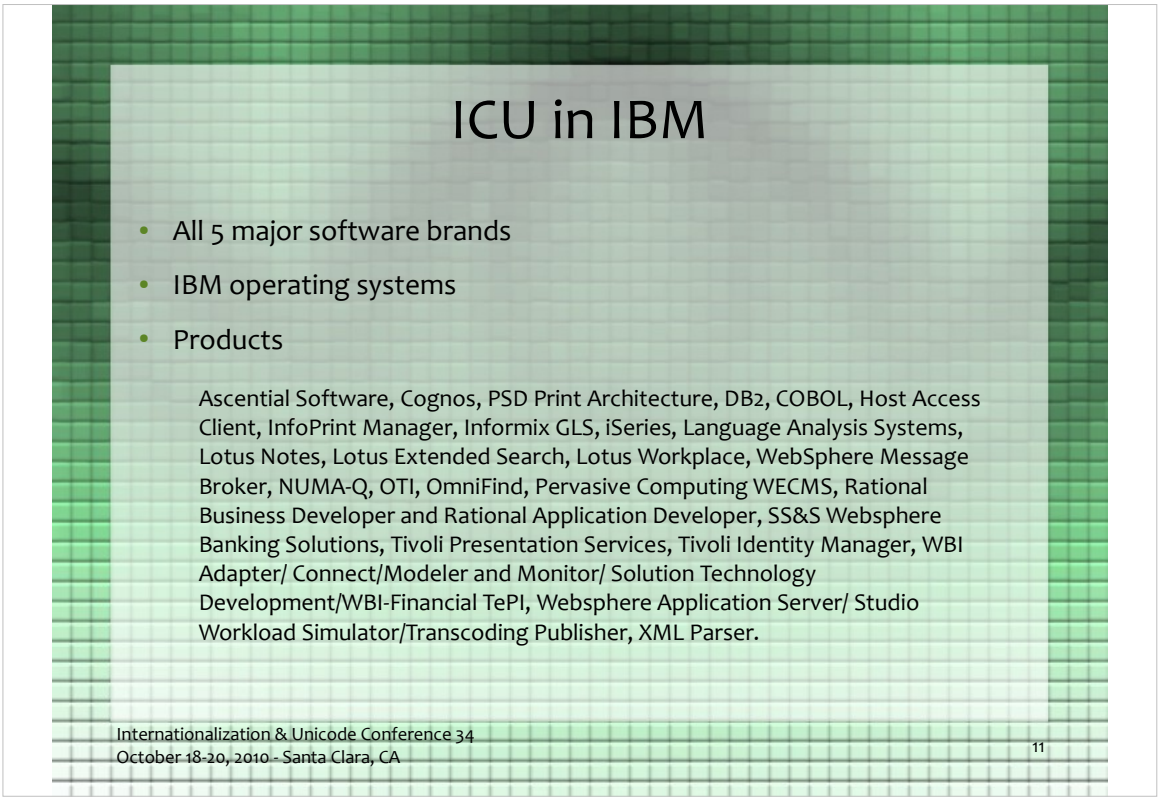
International Components for Unicode (ICU) is a mature set of widely used C/C++ and Java libraries. They are portable to many environments and platforms. There are 3 sub-projects of ICU. There is ICU4C, which is written in C and C++. There is ICU4J which is written in Java.

Mature: celebrated 10 years this year.

ICU is distributed under the X license. The license allows ICU to be incorporated into a wide variety of software projects using the GPL license, while also allowing ICU to be incorporated into non-open source products. You can read the license on ICU's web site for details.

Where is ICU?

- ICU in IBM and Google
- Other ICU Users



ICU is used throughout IBM. It is also used by many other companies and organizations. Many of these companies and organizations also participate in improving ICU.

ICU in Google

- Web Search
- Chrome
- Android
- Adwords
- Google Finance
- Google Maps
- Blogger
- Google Analytics
- Google Gears
- Google Groups
- others...

Other ICU Users

ABAS Software, Adobe, Amazon (Kindle), Amdocs, Apache (Harmony, Lucene, Solr, PDFBox, Tika, Xlan, Xerces, ...), Appian (Mac OS X, iPhone, Safari, iTunes for Windows), Apple, Argonne National Laboratory, Avaya, BAE Systems Geospatial eXploitation Products, BEA, BluePhoenix Solutions, BMC Software, Boost, BroadJump, Business Objects, caris, CERN, Debian Linux, Dell, Eclipse, eBay, EMC Corporation, ESRI, Free BSD, Gentoo Linux, GroundWork Open Source, GTK+, Harman/Becker Automotive Systems GmbH, HP, Hyperion, IBM, Inktomi, Innodata Isogen, Informatica, Intel, Interlogics, IONA, IXOS, Jikes, Library of Congress, Mathworks, Mozilla, Netezza, OpenOffice, Lawson Software, Leica Geosystems GIS & Mapping LLC, Mandrake Linux, OCLC, Progress Software, Python, QNX, Rogue Wave, SAP, SIL, SPSS, Software AG, Sun Microsystems (Solaris, Java), SuSE, Sybase, Symantec, Teradata (NCR), Trend Micro, Virage, webMethods, Wine, WMS Gaming, XyEnterprise, Yahoo!, and many others.

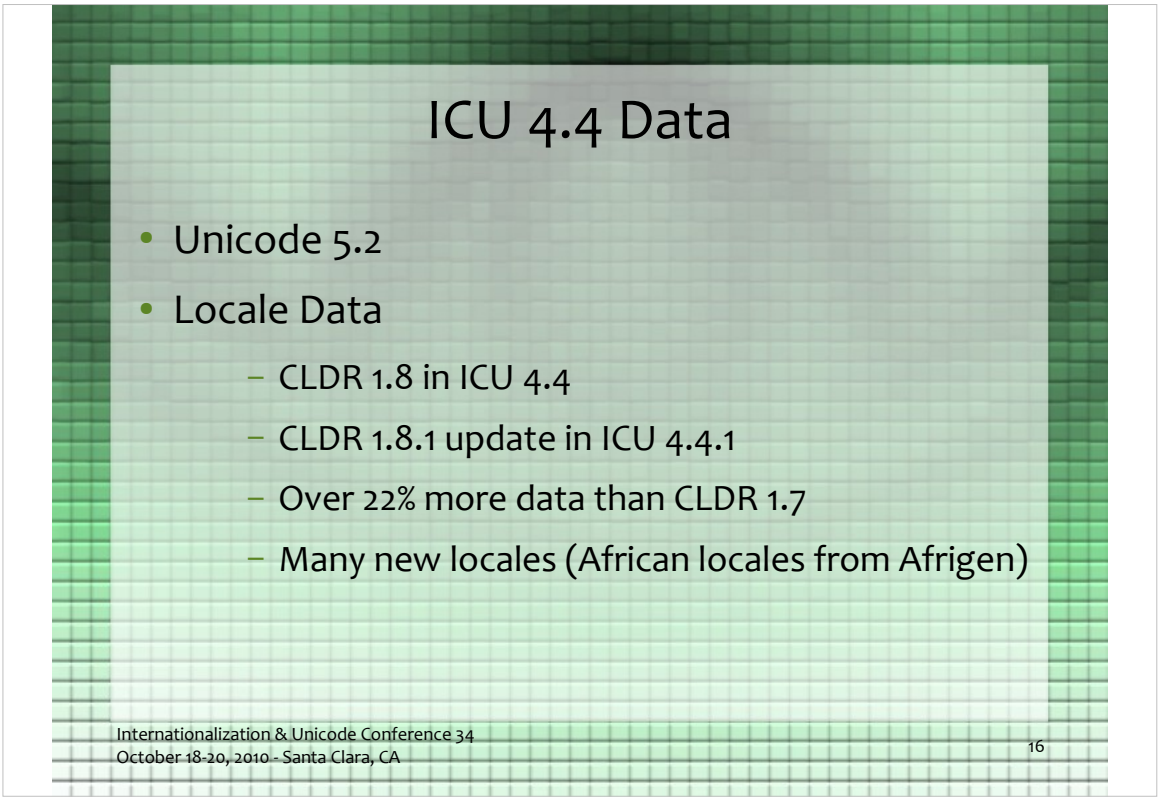
What's new in ICU 4.4?

- Release Summary
- Normalizer2
- Resource Bundle Optimization
- Big Decimal Support (C)
- Select Format
- 64bit Time Zone Transitions
- Java 5 Syntax
- ICU4J source package structure

Release Summary

http://icu-project.org/download/4.4.html

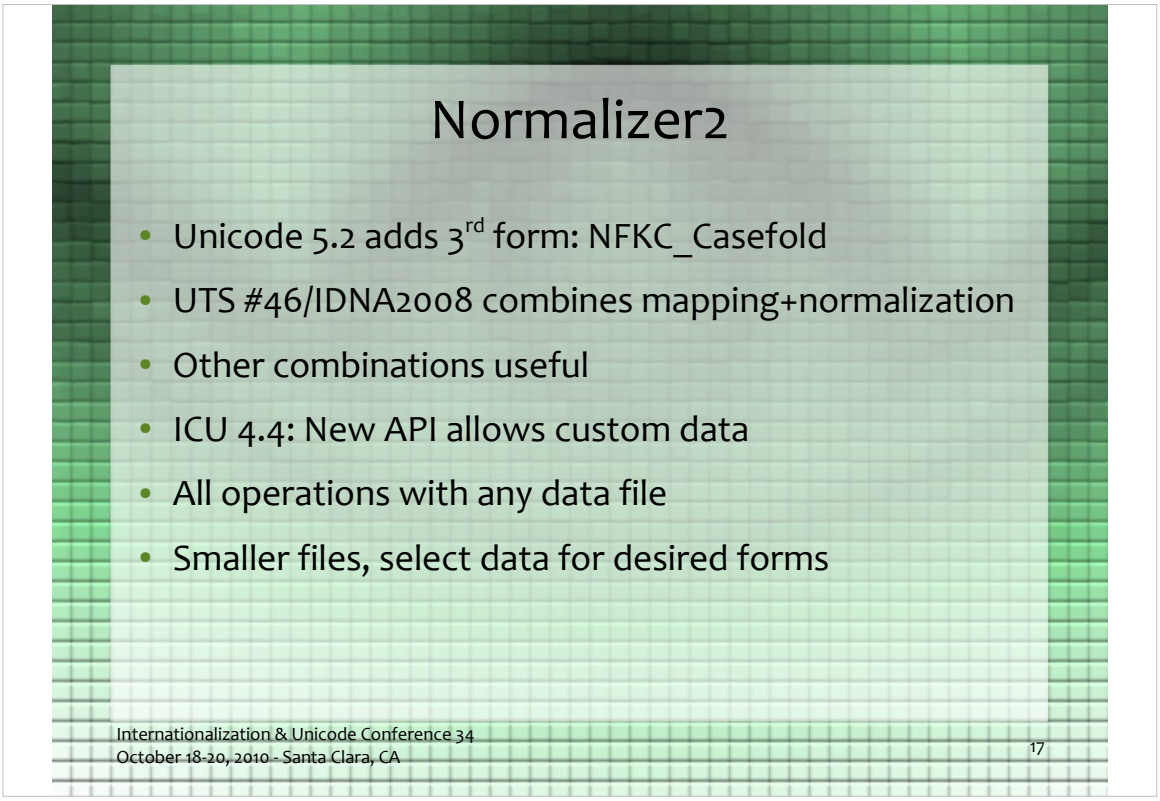
- 2010-03-17 ICU 4.4 (C & J)
- 2010-04-28 ICU4.4.1 (C & J)
- 2010-06-23 ICU4J 4.4.1.1



Every ICU release implements the current Unicode version, collation and CLDR locale data.

Most of the data size increase in ICU 4.4 is absorbed by a more efficient file data structure (see later slide).

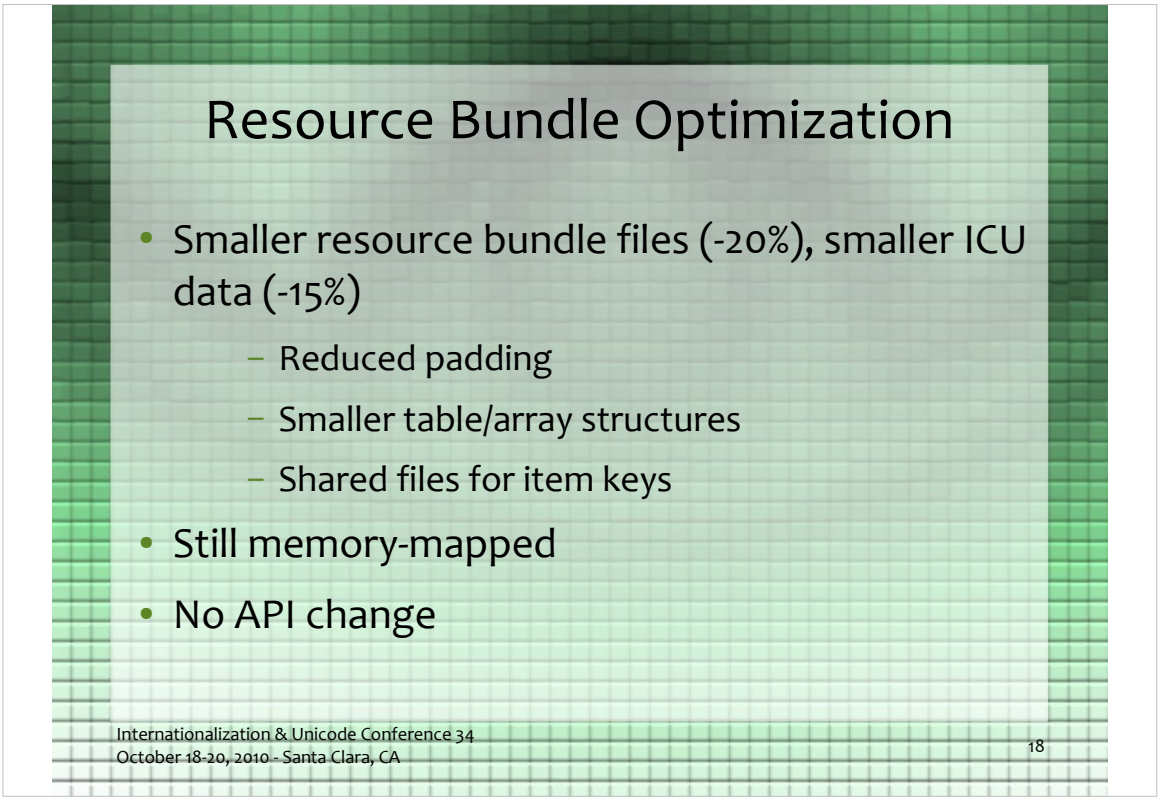
The main locale data file set was split for modularization (allowing selection of less data for selected features). Display names for languages, regions, time zones and currencies are organized into separate file sets in the ICU data package.



Unicode 5.2 defines the NFKC_Casefold combination of character mapping and normalization, and a similar combination can be used to implement UTS #46/IDNA2008. Other mapping+normalization combinations can be useful.

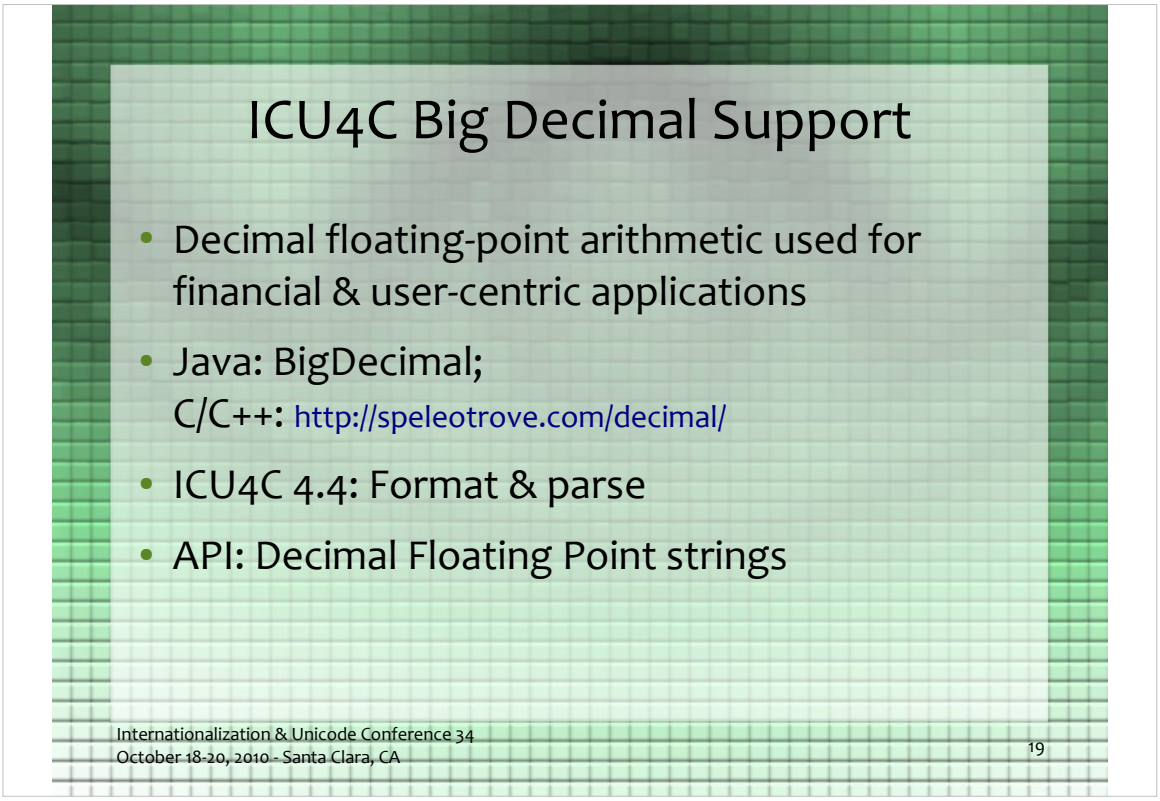
ICU 4.4 provides new normalization API that supports all standard forms and allows for custom data files. All operations are available with any data file, including FCD and FCC processing (see http://www.unicode.org/notes/tn5/) and finding normalization boundaries.

The one data was split into two smaller ones (plus one for NFKC_Casefold).



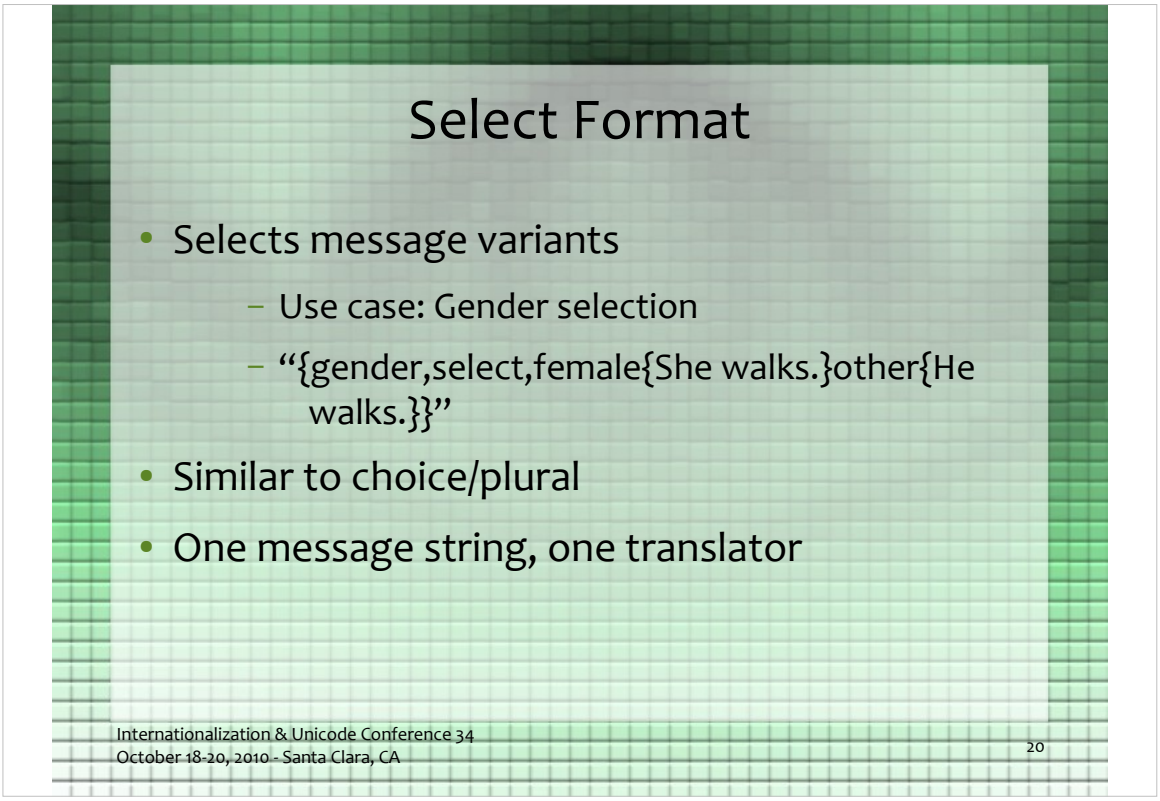
A large portion of ICU's data consists of locale data strings. ICU 4.4 optimizes the resource bundle data structure, yielding significant size reductions while maintaining memory-mapping (no heap usage).

Compressing strings would have resulted in even smaller files at the cost of much increased heap memory usage and time to decompress strings.



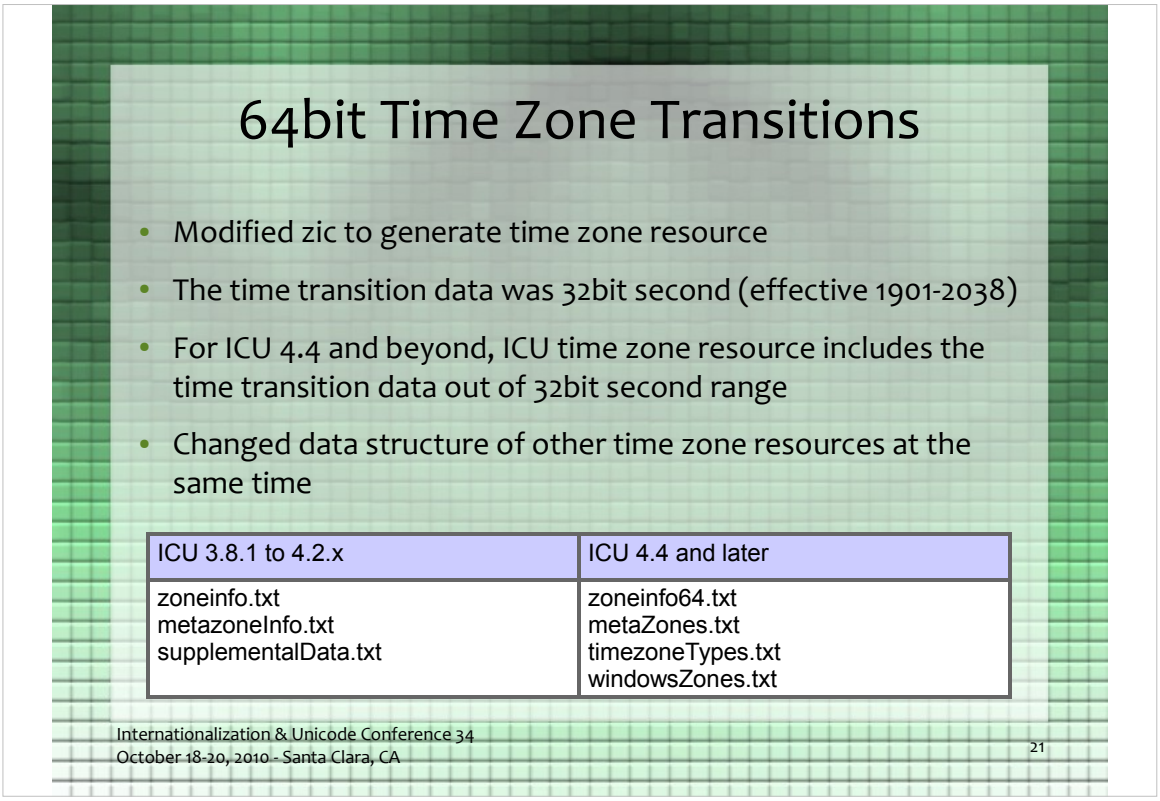
ICU4C 4.4 adds support for formatting and parsing of decimal floating point values equivalent to Java BigDecimal. These are important for high-precision handling of monetary values, and for precise, user-friendly decimal fractions.

ICU does not expose the decNumber code as public API because decNumber's many build options make its structs incompatible between different versions.



The new SelectFormat and "select" argument type in MessageFormat patterns support keyword-based selection of message variants. This is intended to be used for gender selection. Like with choice and plural formats, minimal variant fragments can be selected, but selecting whole-sentence variants is easier to translate.

Combining all variants into one message string can help with translation consistency: Otherwise different translators might translate the variants quite differently.

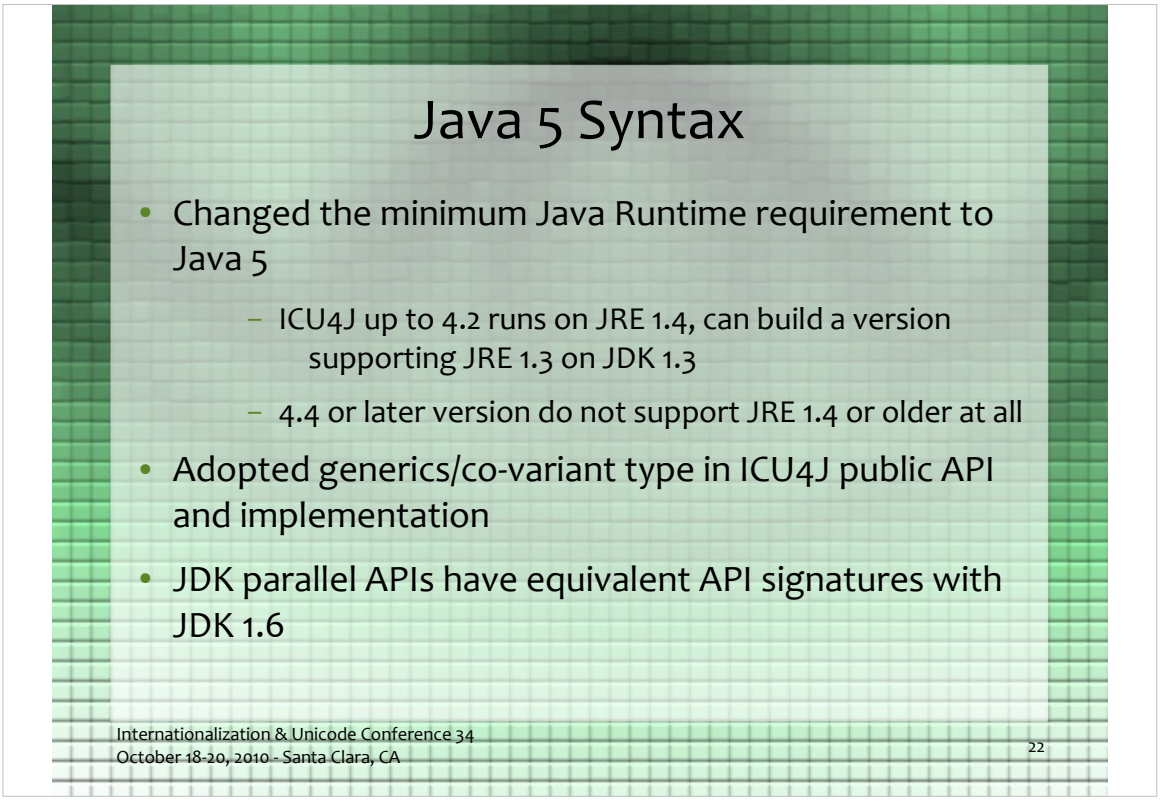


ICU is the consumer of the tz database, known as Olson time zone. "zic" is the tool used for compiling the time zone source file into binary runtime format.

"zic" generates time zone transition data in two formats – one uses 32bit second and another uses 64bit second

There are transitions defined by the tz database before 32bit second range. (With 32bit seconds, 1901-12-13 20:45:52 – 2038-01-19 03:14:07)

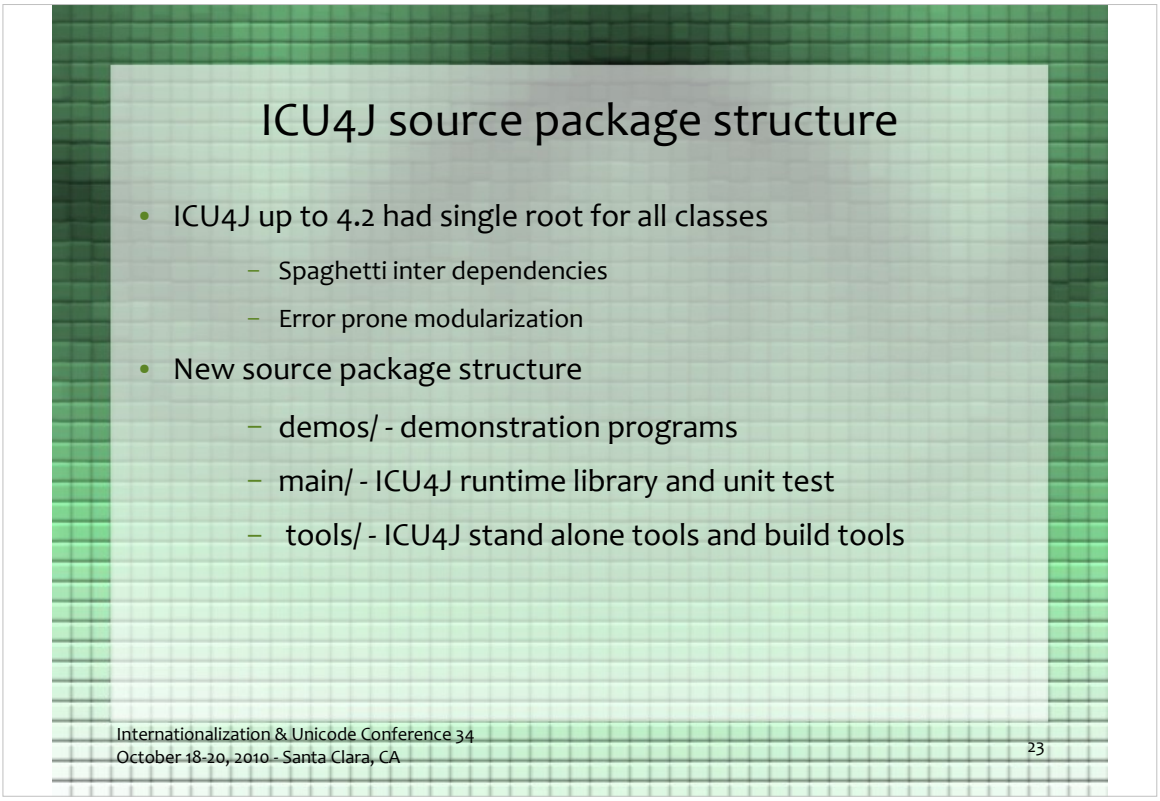
Also isolate time zone related resources from others. Previously, supplementalData contained other type of resources. That will make us easier to keep the same set of data for ICU 4.4 and beyond.



ICU4J no longer support build option producing library works on JRE1.4 or older.

Not 100% binary compatible, that means, existing ICU4J consumer code need to be recompiled with the new ICU4J version.

Practically, 100% source code compatible with really rare exceptions – some API signature changes for co-variant type support may require an update.



For better code/data modularization

ICU4J source package structure (cont.)

Structure under main/

```
classes/
        charset/        Implementation of Java charset service provider
        collate/        Collation service and its dependant
        core/           Unicode, formatting and other miscellaneous services
        currdata/       Currency localized name provider
        langdata/       Language localized name provider
        localespi/      Implementation of Java Locale service provider
        regiondata/     Region localized name provider
        translit/       Transliterator service and its dependant
shared/
        build/          Common component build scripts and properties
        data/           ICU data imported from ICU4C build
        license/        License files
tests/
        charset/        Test cases for charset service
        collate         Test cases for collation service
        core/           Test cases for Unicode, formatting and other services
        framework/      Common test framework and utilities
        localespi/      Test cases for locale SPI
        packaging/      Test cases for validating packaging without localized name providers
        translit/       Test cases for transliterator service
```


Each subcomponent creates its own output jar internally.

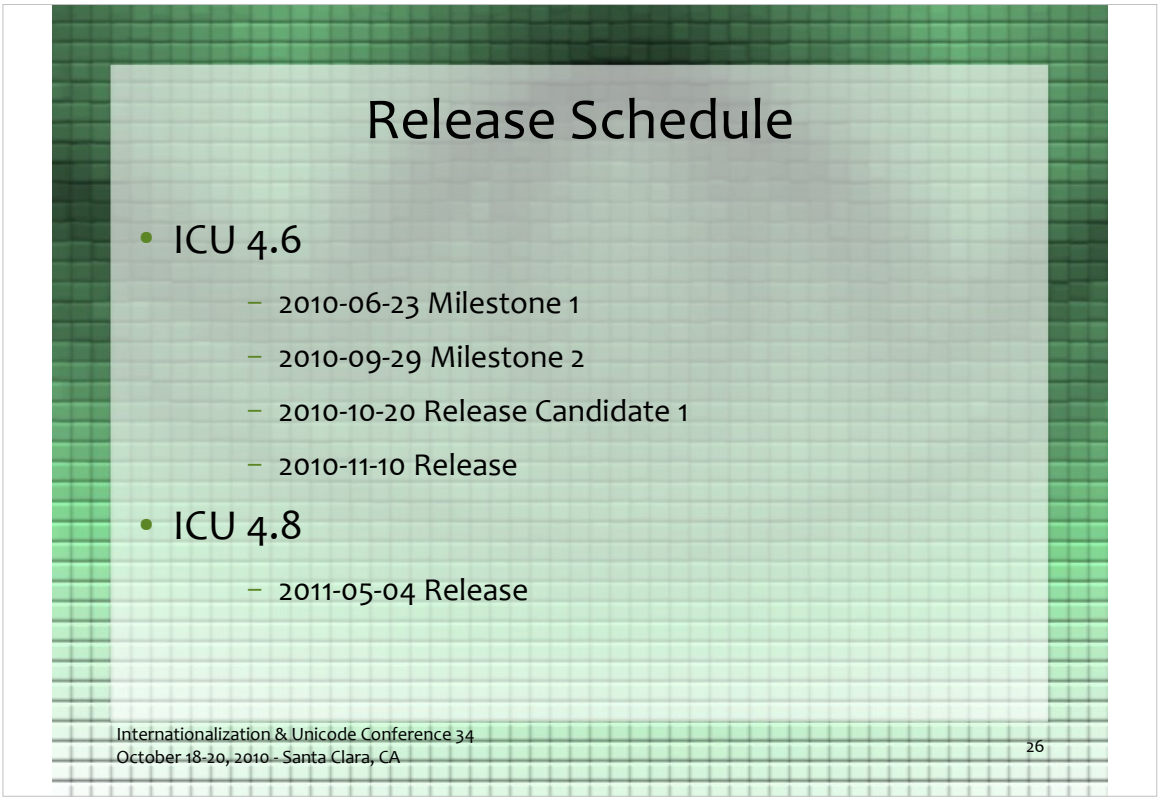
currdata/langdata/localedata/regiondata contain data with small stub code. “core” lib can run without them.

“core” might be refactored and separated into smaller common code and others.

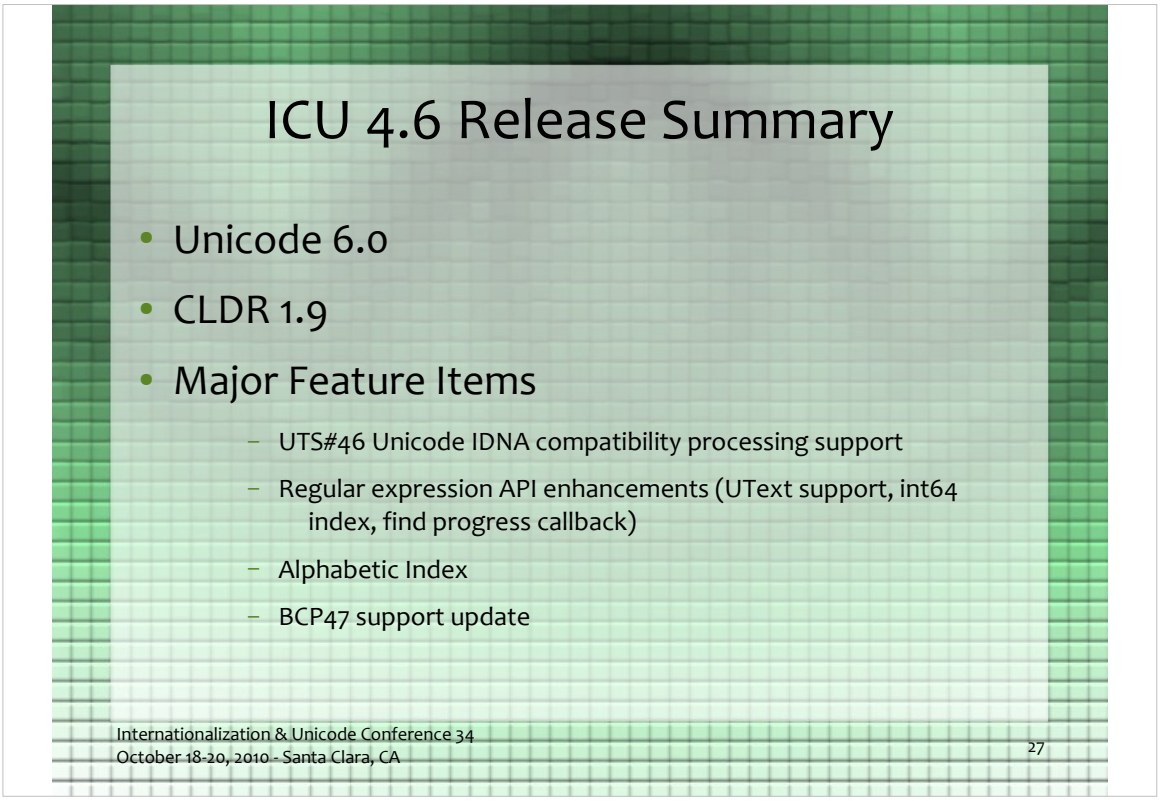
As of ICU 4.4, we still publish all-in-one ICU4J jar. However, when this internal refactoring is once completed, we may also offer smaller independent jars.

What's next for ICU?

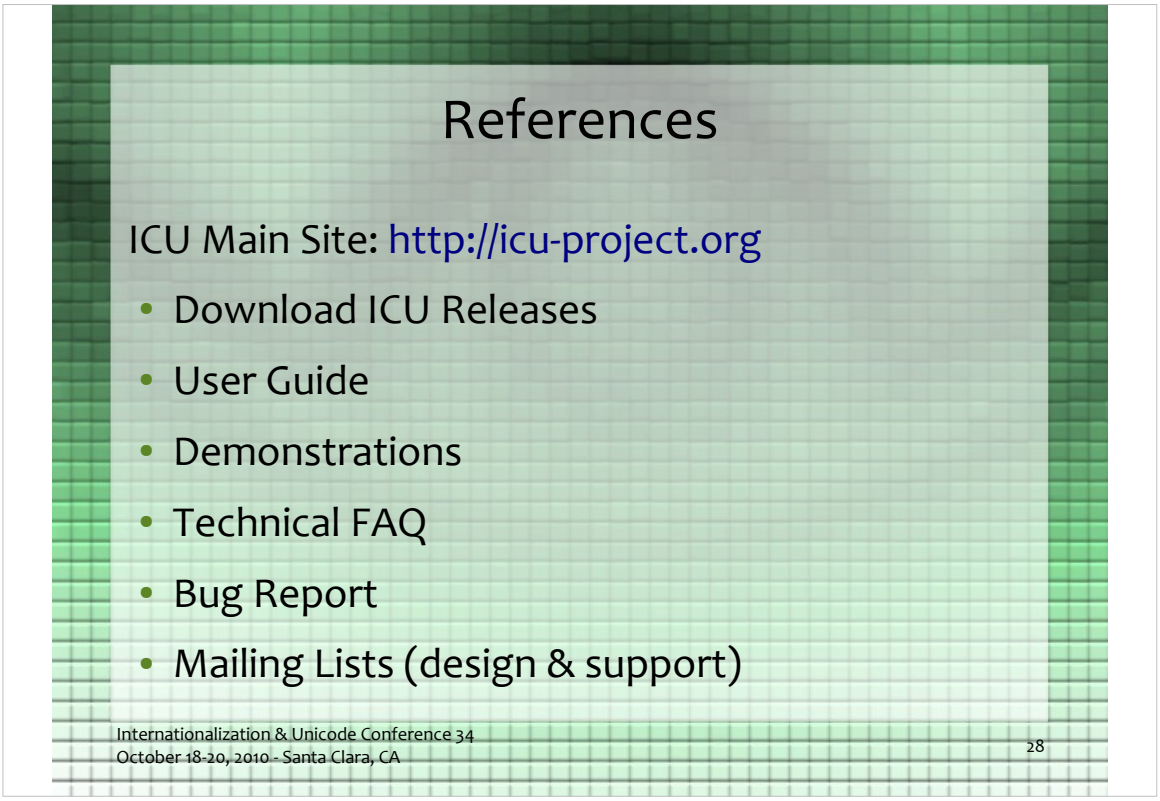
- Release Schedule
- ICU 4.6 Release Summary



ICU4.6 release is synchronized with CLDR 1.9 release.
ICU4.8 is synchronized with CLDR 2.0 release.



CLDR 1.9 is a short cycle release and not getting locale data updates from the survey tool. Therefore, you do not see much locale data difference in ICU except for collation data.



If you would like more information about ICU, you can go to our main site on http://icu-project.org There you will find links to download ICU, the ICU User Guide, the technical FAQ, where to get support, demonstrations of how ICU works, and many other topics related to ICUYou can also find more information about Unicode at the unicode.org site.