

Why ICU4J in the Eclipse Frameworks?

Executive Summary

The Internet has changed the perception and expectations of technology around the world. Whereas in the past users could be expected to learn a few English commands or put up with certain idiosyncrasies (such as having a date expressed in the mm/dd/yy format), they now expect applications to interface with them in their language of choice, and process and display information in a way that is culturally acceptable to them. They know that there are many choices and they no longer have to tolerate applications or Websites that do not respect their needs.

Eclipse aspires to be the premier development platform worldwide, and therefore it should provide developers with the most comprehensive globalization support in the industry and the tools they need to create applications that will work for users in different parts of the world. For Java applications, the Eclipse projects have traditionally depended on the JRE to provide the necessary classes and other development information needed to handle different languages. But Java has been slow to provide comprehensive and accurate information and has fallen far behind Windows and .Net as a platform for developing usable applications for significant parts of the world.

To make up for the deficiencies in Java and to provide best-in-class globalization support, it is proposed that the Eclipse frameworks include ICU4J (International Components for Unicode for Java¹) in its distribution and that the Eclipse open source projects make modest changes to exploit ICU4J's capabilities. ICU4J is an open source library already used by many software vendors and recognized as the most complete and accurate globalization library available in the market² (see Appendix 2 for a list of companies that are using ICU).

What users want

Before discussing why ICU4J is important to the Eclipse frameworks, it may be useful to discuss why it is important for Eclipse to facilitate the creation of global products.

There is no doubt that the world is getting flatter. Information technology is no longer limited to the developed world. Internet users outside of North America now constitute 75% of total users. This group is growing at a much faster rate³ than its North American counterpart. The redistribution of wealth and skills worldwide is also creating a growing consumer class outside of the English speaking world that will be the driving force for Internet commerce in the next decades⁴.

Many of these new users do not speak English, and many of those who do speak English prefer to work with applications and Websites in their native language. They can afford to be more demanding because they are more sophisticated and because they now have more choices. Not only do they expect the information to be in their preferred language, they also expect information to be presented in the expected format. Displaying January 3, 2005 as

(mm/dd/yy)

¹ See <http://www-306.ibm.com/software/globalization/icu/index.jsp>

² See, for example, <http://www.cl.cam.ac.uk/~mgk25/unicode.html> by Markus Kuhn or <http://portal.acm.org/citation.cfm?id=515331> by Richard Gillam

³ There are a number of websites that provide statistics on Internet users, one of them is at <http://www.internetworldstats.com/stats2.htm>

⁴ See, for example, Friedman, Thomas, *The World is Flat*, Farrar, Straus and Giroux, 2005

and expecting the end user to perform the mental gymnastics to convert it was acceptable once, but not anymore. Other considerations include display of numbers and currency, correct spelling for the days of the week, correct collation of data and correct parsing of data entered.

The Internet forces competitors from all over the world to compete for the same set of customers. Not so long ago, a user in China would interact with a program hosted on a server in China, and the applications running on it were either developed elsewhere and then enabled and translated into Chinese or developed locally in Chinese. Now this user can access applications developed and hosted all over the world. This massive migration over the past twenty years has resulted in pockets of communities in many countries where the language of choice is something other than the accepted national language of that country. A company that operates only in a single country may still decide to cater to such communities by providing them with applications in their own language. For example, the U.S. Website of at least one car manufacturer now provides information in English, Spanish, and Chinese.

Finally, users should not be penalized for wanting to see information in their language of choice presented in culturally acceptable formats. Applications should have comparable performance regardless of the language of the data.

What developers need

For most developers, creating applications that will meet the needs of users worldwide is unfamiliar territory, so they depend on their development environment to help them.

Even those familiar with technical issues depend on the platforms or external libraries they use to provide the information specific to a particular locale (a locale here is defined as a language-territory combination such as en-US – English as spoken in the U.S.) since it is unrealistic to expect any one person to know all the most current information for every locale. This is especially important for teams or in distributed development environments to ensure consistency across multiple applications or multiple modules of one application. To help ensure such consistency, the Unicode Consortium launched the Common Locale Data Repository (CLDR)⁵ which provides information for over 200 locales. CLDR's information comes from a number of computer vendors and—since it is open for general use—is easily vetted by sources in various countries.

CLDR provides information in XML files that needs to be transformed into something usable by developers—generally as callable classes. In order for the developers to develop and test such applications, it is crucial that the platforms (or external libraries) they are using provide such classes.

Isn't Java sufficient?

Java provides some powerful facilities for creating global applications. It uses Unicode throughout and locale classes are included as part of the standard. Java 5.0 has some major improvements over Java 1.4.

But as a platform, Java provides support for only a small set of locales (only 21 locales are officially supported, another 73 are shipped but not tested⁶). By comparison, Windows XP SP2 supports 159 locales. In the set of 94 Java locales there is support for only one of the 22 official languages of India, one of the fastest growing economies in the world and a powerhouse for information technology. Appendix 4 contains the lists of locales supported by Windows XP SP2, Windows CE 5.0, Java, ICU4J, MacOS and RedHat.

⁵ See <http://www.unicode.org/cldr>

⁶ See <http://java.sun.com/j2se/1.5.0/docs/guide/intl/locale.doc.html>

In addition, there are some common business practices not supported by Java. For example, it is standard practice in Germany to enter a date based on week number, such as “Wednesday of the 23rd week of the year”, and this is not supported by Java.

As another example, sorting data in a linguistically correct way is an important part of creating global applications. Even though Sun Microsystems is one of the founders of CLDR, Java 5.0 does not use CLDR data and does not pass CLDR conformance testing. The conformance test revealed a number of discrepancies in Java, especially in data collation. One such example, for Japanese data, is shown in Appendix 1.

Java will improve over time. But major releases of Java come out only once every two to three years and bug fixes every nine to twelve months. It will be several years before Java can catch up with what ICU4J offers today. The long release cycle means that Java will not be able to react to rapidly emerging markets. By comparison, ICU4J delivers a new release approximately every nine months. It will continue to improve and can add new locales quickly based on changing market conditions.

Why should ICU4J be shipped as part of the Eclipse Platform?

The development team working on ICU4J originally contributed it to Java. But the team has gone on to significantly improve ICU4J while Java development has not kept up in this particular area. Having ICU4J included in the Eclipse Platform distribution and having the Eclipse projects use ICU4J will result in a number of added advantages for the developers they serve.

Firstly, the state-of-the-art globalization support provided by ICU4J will appeal to non-English speaking users—they will simply have a better experience when they use the Eclipse framework in their native language.

- ICU4J supports and ships twice as many (over 230) locales as Java
- ICU4J uses CLDR as a primary source so it is in full conformance for its globalization classes
- There are additional functions provided by ICU4J not available in Java 5.0, including character normalization, enhanced date and currency formatting, and calendars used in different parts of the world
- For sorting and searching Unicode data, ICU4J can be up to three times faster than Java

Secondly, the ICU4J support in the Eclipse projects will appeal to plug-in developers because they can choose to make use of premier globalization support for their own applications out-of-the-box.

Thirdly, using ICU4J in the Eclipse frameworks means that globalization can be separated from Java, which will help embedded developers who typically lag behind in the versions of Java they are required to use. ICU4J works with both Java 1.4 and Java 5.0 so even applications designed for Java 1.4 can take advantage of its features.

Lastly, it will help maintain the competitiveness of Eclipse as a development environment. This is particularly true given that one of the major enhancements in Microsoft’s Whidbey release is its facilities for developing global applications.⁷

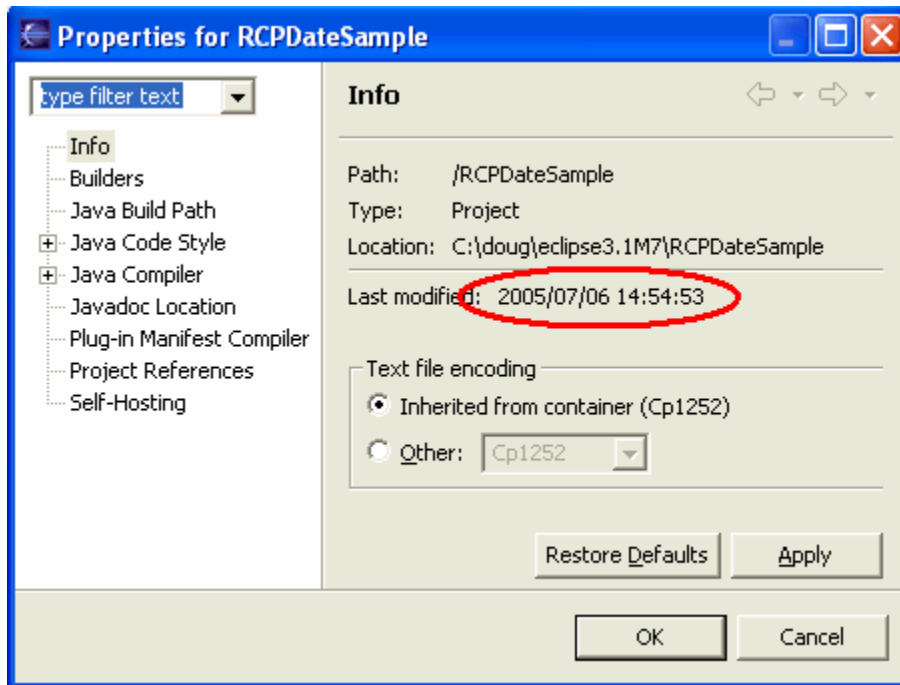
And what changes would the Eclipse projects need to make to support ICU4J?

The code in the Eclipse projects can be modified to call ICU4J with only minor changes. By modifying this code, any application based on the Eclipse frameworks will automatically use the improved globalization support provided by ICU4J. Most code can be changed by either changing package names or substituting a Java class with an ICU4J class with the same signature. In order for the developers to use the Eclipse

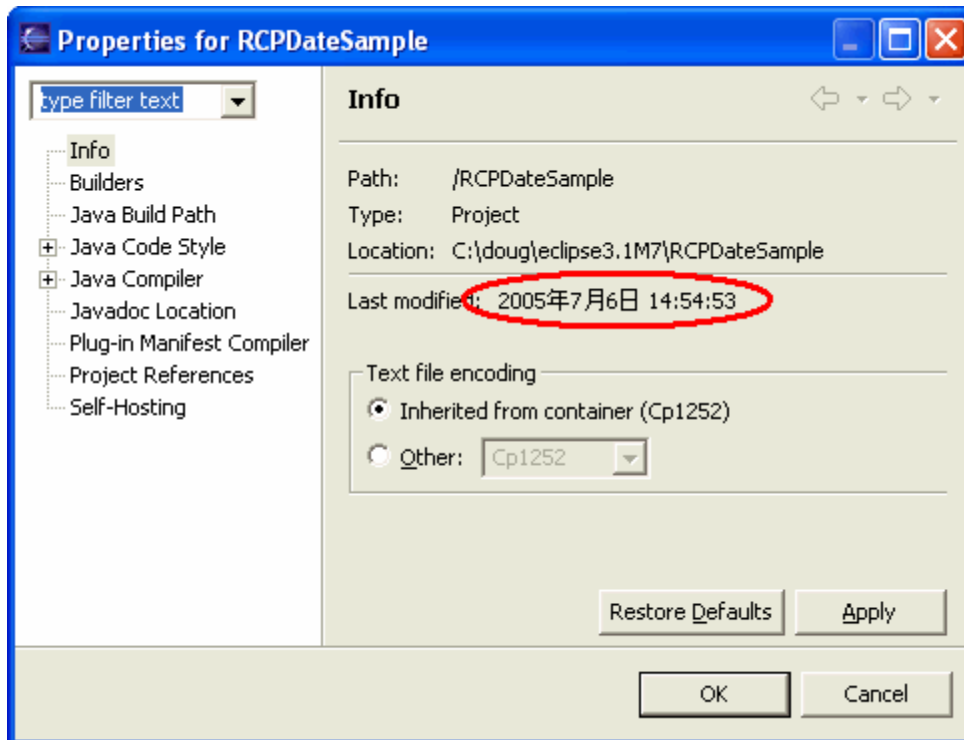
⁷ See *ASP.NET 2.0 Localization Features: A Fresh Approach to Localizing Web Applications* at <http://msdn.microsoft.com/asp.net/beta2/infrastructure/default.aspx?pull=/library/en-us/dnvs05/html/asp2local.asp>

frameworks as an underlying platform, the Eclipse projects will also need to call ICU4J for international support. The following is an example of a simple change to one Eclipse component resulting in a more culturally acceptable interface.

The standard Java DateFormat class returns a generic date and time when the locale is set for Japanese. This is used in a number of places across the Eclipse projects, including the example below.



By changing a few lines in this Eclipse component, a Japanese date, which is more acceptable to Japanese speakers, will be displayed instead as shown below.



The following results from *diff* show the changes required to display the dates in Japanese format.

```
import java.io.File;
-import java.text.DateFormat;
-import java.text.SimpleDateFormat;
+import com.ibm.icu.text.DateFormat;
+import com.ibm.icu.text.SimpleDateFormat;
import java.util.Date;

import org.eclipse.core.resources.IResource;
```

The following is a list of the files that were changed :

```
plugins/org.eclipse.compare/compare/org/eclipse/compare/EditionSelectionDialog.java
plugins/org.eclipse.compare/compare/org/eclipse/compare/internal/AddFromHistoryDialog.java
plugins/org.eclipse.jdt.launching/launching/org/eclipse/jdt/internal/launching/StandardVMRunner.java
plugins/org.eclipse.pde.ui/src/org/eclipse/pde/internal/ui/view/FileAdapterPropertySource.java
plugins/org.eclipse.team.cvs.ui/src/org/eclipse/team/internal/ccvs/ui/CVSFilePropertiesPage.java
plugins/org.eclipse.team.cvs.ui/src/org/eclipse/team/internal/ccvs/ui/HistoryTableProvider.java
plugins/org.eclipse.team.cvs.ui/src/org/eclipse/team/internal/ccvs/ui/subscriber/CVSChangeSetCollector.java
plugins/org.eclipse.team.cvs.ui/src/org/eclipse/team/internal/ccvs/ui/subscriber/OpenChangeSetAction.java
plugins/org.eclipse.team.ui/src/org/eclipse/team/internal/ui/synchronize/SubscriberRefreshSchedule.java
plugins/org.eclipse.text/src/org/eclipse/jface/text/templates/GlobalTemplateVariables.java
plugins/org.eclipse.ui.ide/src/org/eclipse/ui/internal/ide/dialogs/ResourceInfoPage.java
plugins/org.eclipse.ui.ide/src/org/eclipse/ui/views/bookmarkexplorer/MarkerUtil.java
plugins/org.eclipse.ui.ide/src/org/eclipse/ui/views/markers/internal/Util.java
plugins/org.eclipse.ui.ide/src/org/eclipse/ui/views/properties/ResourcePropertySource.java
```

plugins/org.eclipse.ui.ide/src/org/eclipse/ui/views/tasklist/MarkerUtil.java
plugins/org.eclipse.ui.workbench/Eclipse UI/org/eclipse/ui/internal/progress/NewProgressViewer.java
plugins/org.eclipse.update.core/src/org/eclipse/update/core/Utilities.java
plugins/org.eclipse.update.core/src/org/eclipse/update/internal/security/JarVerificationResult.java
plugins/org.eclipse.update.ui/src/org/eclipse/update/internal/ui/views/InstallationHistoryAction.java
plugins/org.eclipse.update.ui/src/org/eclipse/update/internal/ui/wizards/ShowActivitiesDialog.java

In Summary

ICU4J provides the best globalization support available in the software industry today. As Eclipse strives to become the premier development platform worldwide, including support for ICU4J would benefit their users as well as their user's users. In addition, it would position Eclipse to compete and win against Microsoft in an area where Microsoft believes it cannot be beaten. Finally, the changes required in the Eclipse projects are low risk and not difficult.

Appendix 1 – Differences between Java and ICU4J

The left table shows the correct way of sorting a block of Japanese data using ICU4J and conforming to the JIS X4061 standard. The right table shows the same block of data incorrectly sorted using Java collation (the red boxes highlight the incorrect results).

String	Sort Index
シヤール	0/76
シヤイ	1/76
シヤイ	2/76
シヤレ	3/76
ちよこ	4/76
ちよこ	5/76
チョコレート	6/76
てーた	7/76
テータ	8/76
テェタ	9/76
てえた	10/76
でーた	11/76
データ	12/76
デェタ	13/76
でえた	14/76
てーたー	15/76
テータァ	16/76
テェター	17/76
てえたぁ	18/76
てえたー	19/76
でーたー	20/76

String	Sort Index
シヤール	0/76
シヤイ	2/76
シヤイ	1/76
シヤレ	3/76
ちよこ	5/76
ちよこ	4/76
チョコレート	6/76
てえた	10/76
テェタ	9/76
テータ	8/76
でえた	14/76
デェタ	13/76
てえたぁ	18/76
テェター	17/76
テータァ	16/76
でェタァ	22/76
デェタぁ	23/76
デェタァ	24/76
てえたー	19/76
てーた	7/76
でーた	11/76

Code used to create the result above :

```
Collator collator =
    Collator.getInstance(Locale.JAPAN);
collator.setStrength(Collator.QUATERNARY);
viewer.setSorter(new ViewerSorter(collator));
```

Code used to create the result above :

```
Collator collator =
    Collator.getInstance(Locale.JAPAN);
collator.setStrength(Collator.IDENTICAL);
viewer.setSorter(new ViewerSorter(collator));
```

The lack of a sufficient set of tested locale data from Java means that end users are presented with information not in the local language or that is incorrect. The following example shows differences in ICU date formatting (on the left) vs Java date formatting (on the right). The red boxes highlight the locales where the data from Java are either absent or incorrect.

ja_JP	2005年7月6日 23:01:29:PDT	2005/07/06 23:01:29 PDT
kk_KZ	6 шілде 2005 ж. 23:01:29 GMT-07:00	July 6, 2005 11:01:29 PM PDT
kn_IN	6 ಜುಲೈ, 2005 11:01:29 ಅಪರಾಹ್ನ, GMT-...	July 6, 2005 11:01:29 PM PDT
ko_KR	2005년 7월 6일 오후 11시 01분 29초	2005년 7월 6일 (수) 오후 11시 01...
lt_LT	2005 m. liepos 6 d. 23:01:29 GMT-0...	Trečdiena, 2005, Liepos 6 23.01.2...
lv_LV	2005. gada 6. jūlijs 23:01:29 GMT-0...	trešdiena, 2005, 6 jūlijs 23:01:29 PDT
mk_MK	06 јули 2005 23:01:29 GMT-07:00	6, јули 2005 23:01:29 PDT
mr_IN	6 जुलै 2005 11:01:29 म.न. GMT-07:00	July 6, 2005 11:01:29 PM PDT
ms_MY	06 Julai 2005 11:01:29 PM GMT-07:00	July 6, 2005 11:01:29 PM PDT
mt_MT	6 ta' Lulju 2005 23:01:29 GMT-07:00	July 6, 2005 11:01:29 PM PDT
nb_NO	6. juli 2005 23.01.29 GMT-07:00	July 6, 2005 11:01:29 PM PDT
nl_BE	6 juli 2005 23:01:29 PDT	6 juli 2005 23:01:29 PDT
nl_NL	6 juli 2005 23:01:29 PDT	6 juli 2005 23:01:29 PDT
pa_IN	੬ ਜੁਲਾਈ ੨੦੦੫ ੧੧:੦੧:੨੯ ਸ਼ਾਮ GMT-੦੭:੦੦	July 6, 2005 11:01:29 PM PDT

The above example is created using the following code fragments:

```
import com.ibm.icu.text.DateFormat; // replaces import java.text.DateFormat
import com.ibm.icu.util.ULocale; // replaces import java.util.Locale

[...]
```

```
static final Date now = new Date();
static final String dateInLocale(ULocale ulocale) {
    DateFormat fmt = DateFormat.getDateInstance(DateFormat.LONG,
        DateFormat.LONG, ulocale);
    return fmt.format(now);
}
```


Appendix 2 – Companies and Projects that Use ICU

Many companies and projects are using ICU in their products. These companies include:

- Adobe
- Apple (Mac OS X)
- ABAS Software
- Argonne National Laboratory
- Ascential Software
- Avaya
- BEA
- BroadJump
- BluePhoenix Solutions
- BMC Software (Remedy)
- Business Objects
- CARIS
- CERN
- Cognos
- Debian Linux
- Gentoo Linux
- HP
- Hyperion
- IBM
- Inktomi
- Innodata Isogen
- Informatica
- Intel
- Interlogics
- IONA
- IXOS
- Jikes
- Macromedia
- Mathworks
- Mozilla
- OpenOffice
- Language Analysis Systems
- Lawson Software
- Leica Geosystems GIS & Mapping LLC
- Mandrake Linux
- Oracle
- Parrot
- PayPal
- Progress Software
- Python
- QNX
- Rogue Wave
- SAP
- Siebel
- SIL
- SPSS
- Software AG
- SuSE
- Sybase
- Teradata (NCR)
- Trend Micro
- Virage
- webMethods
- Wine
- WMS Gaming

Appendix 3 – Sample products from one company that uses ICU: IBM

- Application & Integration Middleware (AIM), Embedded Platform Development
- Boulder PSD Print Architecture
- COBOL
- Content Management
- DB2 and IMS Tools
- DB2 Entity Analytic Solutions
- DB2 High Performance Unload (Data Management Tools)
- DB2 Multiplatform Tools and Recovery Expert for Multiplatforms
- DB2 UDB
- DB2 Information Integrator
- DB2 Information Integrator Classic Federation for z/OS
- DB2 Information Integrator Event Publisher V8.2
- DB2 Information Integrator Enterprise Search
- DB2 Information Integrator Find It Edition v8.2 (was Enterprise Search)
- DB2 Recovery Expert (Data Management Tools)
- DB2 Warehouse Manager
- eServer Software Development
- eServer Tools Development
- Extended Parallel Server
- Host Access Client
- Host Access Component Development
- IBM Liquidity Manager
- Industry Solutions
- Infoprint Manager
- Infoprint Development z/OS (Printing Systems Division)
- Informix GLS version 4.0
- Install Strategy and Development
- Integrated Runtime
- iSeries
- iSeries Printing
- LanguageWare
- Lotus Notes
- Lotus Extended Search
- Lotus Workplace (LWP)
- MQ integrator Endeavour
- NUMA-Q
- OTI
- Pervasive Computing WECMS
- Rational Application Developer (RAD)
- Rational Eclipse-based offerings
- Rational
- Retail Store Solutions
- SS&S Websphere Banking Solutions
- Systems and Software Development
- Tivoli
- Tivoli Application Management (Cyanea acquisition)
- Tivoli Identity Manager
- Tivoli Monitoring for Transaction Performance
- Tivoli Workload Scheduler
- Tivoli Presentation Services
- TXSeries
- WBI Adapter
- WBI Brokers
- WBI Connect
- WBI Development (WBI for Financial Networks)
- WBI Modeler and Monitor
- WBI Server
- WBI Solution Technology Development
- WBI-Financial TePI
- WebSphere Application Server
- Websphere DataStage
- WebSphere Information Integrator
- WebSphere Information Integrator Content Edition
- WebSphere Information Integrator OmniFind
- WebSphere Partner Gateway
- WebSphere Product Information Management Development
- Websphere Studio Workload Simulator
- Websphere Transcoding Publisher
- Workplace, Portal and Collaboration Software (WPLC)
- WSS Business Process Solutions
- XML Parser
- XSLT4C

Appendix 4 – Comparisons of Supported Locales

This appendix lists locale as well as text entry and display support information for ICU4J 3.4, Java 5.0, Windows XP SP2, Windows CE 5.0, MacOS X 10.4 and RedHat RHEL 4. MacOS 10.3 and later versions use ICU for much of their globalization support.

Please note the following:

- The information in the ICU and Java columns below is based on the locale support provided by these platforms. It highlights their differences—most notably the support included in ICU that is not included in Java. For more details, refer to:
 - For ICU: <http://dev.icu-project.org/cgi-bin/viewcvs.cgi/icu/source/data/locales/>
 - For Java: <http://java.sun.com/j2se/1.5.0/docs/guide/intl/locale.doc.html>
- The information in the Windows XP and Windows CE columns is based on the locale, text entry and display support provided by these platforms. Microsoft sees this extensive list as a competitive advantage.
 - For Windows XP: <http://www.microsoft.com/globaldev/reference/winxp/XPLocLang.mspx> and <http://www.microsoft.com/globaldev/reference/win2k/setup/lcid.mspx>
 - For Windows CE: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceinternational5/html/wce50conlanguageidentifiersandlocales.asp>
- The information in the MacOS and RedHat columns is based on the text entry and display support provided by these platforms.

Extending locale, text entry and display support is strategic for all of the major platforms—and they will continue to do so with every new release.

Key: ✓ - **Supported**
 ♦ - **Provided, but not tested**

Language	Country	ICU	Java	Win XP	Win CE	MacOS	Red Hat
Afrikaans	South Africa	✓		✓	✓	✓	✓
Albanian	Albania	✓	♦	✓	✓	✓	
Amharic	Ethiopia	✓					
Arabic	Algeria	✓	♦	✓	✓	✓	✓
	Bahrain	✓	♦	✓	✓	✓	✓
	Egypt	✓	♦	✓	✓	✓	✓
	Iraq	✓	♦	✓	✓	✓	✓
	Jordan	✓	♦	✓	✓	✓	✓
	Kuwait	✓	♦	✓	✓	✓	✓
	Lebanon	✓	♦	✓	✓	✓	✓
	Libya	✓	♦	✓	✓	✓	✓
	Morocco	✓	♦	✓	✓	✓	✓
	Oman	✓	♦	✓	✓	✓	✓
	Qatar	✓	♦	✓	✓	✓	✓
	Saudi Arabia	✓	✓	✓	✓	✓	✓
	Sudan	✓	♦			✓	✓
Syria	✓	♦	✓	✓	✓	✓	

Language	Country	ICU	Java	Win XP	Win CE	MacOS	Red Hat
	Tunisia	✓	♦	✓	✓	✓	✓
	U.A.E.	✓	♦	✓	✓	✓	✓
	Yemen	✓	♦	✓	✓	✓	✓
Armenian	Armenia	✓		✓	✓	✓	
Assamese	India	✓			✓		
Azerbaijani (Cyrillic)	Azerbaijan	✓		✓	✓	✓	
Azerbaijani (Latin)	Azerbaijan	✓		✓	✓	✓	
Basque	Spain	✓		✓	✓	✓	✓
Belarusian	Belarus	✓	♦	✓	✓	✓	
Bengali	India	✓		✓	✓		✓
Bosnian	Bosnia and Herzegovina			✓			✓
Bulgarian	Bulgaria	✓	♦	✓	✓	✓	✓
Catalan	Spain	✓	♦	✓	✓	✓	✓
Chinese (Simplified)	People's Republic of China	✓	✓	✓	✓	✓	✓
	Singapore	✓		✓	✓	✓	✓
	Macao SAR	✓		✓	✓	✓	✓
Chinese (Traditional)	Taiwan	✓	✓	✓	✓	✓	✓
	Hong Kong SAR	✓	♦	✓	✓	✓	✓
Cornish	United Kingdom	✓				✓	
Croatian	Bosnia and Herzegovina			✓		✓	✓
	Croatia	✓	♦	✓	✓	✓	✓
Czech	Czech Republic	✓	♦	✓	✓	✓	✓
Danish	Denmark	✓	♦	✓	✓	✓	✓
Divehi	Maldives			✓	✓		
Dutch	Belgium	✓	♦	✓	✓	✓	✓
	Netherlands	✓	✓	✓	✓	✓	✓
English	Australia	✓	✓	✓	✓	✓	✓
	Belize	✓		✓	✓	✓	✓
	Botswana	✓				✓	✓
	Canada	✓	✓	✓	✓	✓	✓
	Caribbean			✓	✓	✓	✓
	Hong Kong SAR	✓				✓	✓
	India	✓	♦			✓	✓
	Ireland	✓	♦	✓	✓	✓	✓
	Jamaica			✓	✓	✓	✓
	Malta	✓				✓	✓
	New Zealand	✓	♦	✓	✓	✓	✓
	Pakistan	✓				✓	✓
	Philippines	✓		✓	✓	✓	✓
	Singapore	✓				✓	✓
	South Africa	✓	♦	✓	✓	✓	✓
Trinidad				✓	✓	✓	✓

Language	Country	ICU	Java	Win XP	Win CE	MacOS	Red Hat
	United Kingdom	✓	✓	✓	✓	✓	✓
	United States	✓	✓	✓	✓	✓	✓
	Virgin Islands (US)	✓				✓	✓
	Zimbabwe	✓		✓	✓	✓	✓
Estonian	Estonia	✓	♦	✓	✓	✓	✓
Faroese	Faeroe Islands	✓		✓	✓	✓	
Finnish	Finland	✓	♦	✓	✓	✓	✓
French	Belgium	✓	♦	✓	✓	✓	✓
	Canada	✓	✓	✓	✓	✓	✓
	France	✓	✓	✓	✓	✓	✓
	Luxembourg	✓	♦	✓	✓	✓	✓
	Monaco	✓		✓	✓	✓	✓
	Switzerland	✓	♦	✓	✓	✓	✓
Gallegan	Spain	✓		✓	✓	✓	
Georgian	Georgia			✓	✓		
German	Austria	✓	♦	✓	✓	✓	✓
	Belgium	✓				✓	✓
	Germany	✓	✓	✓	✓	✓	✓
	Liechtenstein	✓		✓	✓	✓	✓
	Luxembourg	✓	♦	✓	✓	✓	✓
	Switzerland	✓	♦	✓	✓	✓	✓
Greek	Greece	✓	♦	✓	✓	✓	✓
Gujarati	India	✓		✓	✓	✓	✓
Hebrew	Israel	✓	✓	✓	✓	✓	✓
Hindi	India	✓	✓	✓	✓	✓	✓
Hungarian	Hungary	✓	♦	✓	✓	✓	✓
Icelandic	Iceland	✓	♦	✓	✓	✓	✓
Inari Sami	Finland			✓			
Indonesian	Indonesia	✓		✓	✓	✓	
Irish	Ireland	✓				✓	
Italian	Italy	✓	✓	✓	✓	✓	✓
	Switzerland	✓	♦	✓	✓	✓	✓
Japanese	Japan	✓	✓	✓	✓	✓	✓
Kalaallisut	Greenland	✓				✓	
Kannada	India	✓		✓	✓		
Kazakh	Kazakhstan	✓		✓	✓	✓	
Kirghiz	Kyrgyzstan			✓	✓		
Konkani	India	✓		✓	✓		
Korean	Korea	✓	✓	✓	✓	✓	✓
Latvian	Latvia	✓	♦	✓	✓	✓	
Lithuanian	Lithuania	✓	♦	✓	✓	✓	
Lule Sami	Norway			✓			
	Sweden			✓			
Macedonian	Macedonia	✓	♦	✓	✓	✓	✓
Malay	Brunei Darussalam	✓		✓	✓	✓	✓
	Malaysia	✓		✓	✓	✓	✓
Malayalam	India	✓		✓	✓		

Language	Country	ICU	Java	Win XP	Win CE	MacOS	Red Hat
Maltese	Malta	✓		✓		✓	
Manx	United Kingdom	✓				✓	
Maori	New Zealand			✓			✓
Marathi	India	✓		✓	✓	✓	✓
Mongolian	Mongolia			✓	✓		
Northern Sami	Finland			✓			
	Norway			✓			
	Sweden			✓			
Norwegian Bokmål	Norway	✓	♦	✓	✓	✓	✓
Norwegian Nynorsk	Norway	✓	♦	✓	✓	✓	✓
Oriya	India	✓			✓		
Oromo	Ethiopia	✓				✓	
	Kenya	✓				✓	
Pashto (Pushto)	Afghanistan	✓				✓	
Persian	Afghanistan	✓				✓	✓
	Iran	✓		✓	✓	✓	✓
Polish	Poland	✓	♦	✓	✓	✓	✓
Portuguese	Brazil	✓	✓	✓	✓	✓	✓
	Portugal	✓	♦	✓	✓	✓	✓
Punjabi	India	✓		✓	✓	✓	✓
Quechua	Bolivia			✓			
	Ecuador			✓			
	Peru			✓			
Romanian	Romania	✓	♦	✓	✓	✓	✓
Russian	Russia	✓	♦	✓	✓	✓	✓
	Ukraine	✓				✓	✓
Sanskrit	India			✓	✓		
Serbian (Cyrillic)	Bosnia and Herzegovina			✓		✓	
	Serbia	✓		✓	✓	✓	
Serbian (Latin)	Bosnia and Herzegovina			✓		✓	
	Serbia	✓		✓	✓	✓	
Skolt Sami	Finland			✓			
Slovak	Slovakia	✓	♦	✓	✓	✓	✓
Slovenian	Slovenia	✓	♦	✓	✓	✓	✓
Somali	Djibouti	✓				✓	
	Ethiopia	✓				✓	
	Kenya	✓				✓	
	Somalia	✓				✓	
Southern Sami	Norway			✓			
	Sweden			✓			

Language	Country	ICU	Java	Win XP	Win CE	MacOS	Red Hat
Spanish	Argentina	✓	♦	✓	✓	✓	✓
	Bolivia	✓	♦	✓	✓	✓	✓
	Chile	✓	♦	✓	✓	✓	✓
	Colombia	✓	♦	✓	✓	✓	✓
	Costa Rica	✓	♦	✓	✓	✓	✓
	Dominican Republic	✓	♦	✓	✓	✓	✓
	Ecuador	✓	♦	✓	✓	✓	✓
	El Salvador	✓	♦	✓	✓	✓	✓
	Guatemala	✓	♦	✓	✓	✓	✓
	Honduras	✓	♦	✓	✓	✓	✓
	Mexico	✓	♦	✓	✓	✓	✓
	Nicaragua	✓	♦	✓	✓	✓	✓
	Panama	✓	♦	✓	✓	✓	✓
	Paraguay	✓	♦	✓	✓	✓	✓
	Peru	✓	♦	✓	✓	✓	✓
	Puerto Rico	✓	♦	✓	✓	✓	✓
	Spain	✓	✓	✓	✓	✓	✓
	United States	✓				✓	✓
	Uruguay	✓	♦	✓	✓	✓	✓
	Venezuela	✓	♦	✓	✓	✓	✓
Swahili	Kenya	✓		✓	✓	✓	✓
	Tanzania	✓				✓	✓
Swedish	Finland	✓		✓	✓	✓	✓
	Sweden	✓	✓	✓	✓	✓	✓
Syriac	Syria			✓	✓		
Tamil	India	✓		✓	✓	✓	✓
Tatar	Tatarstan			✓	✓		
Telugu	India	✓		✓	✓		
Thai	Thailand	✓	✓	✓	✓	✓	
Tigrinya	Eritrea	✓					
	Ethiopia	✓					
Tswana	South Africa			✓			
Turkish	Turkey	✓	♦	✓	✓	✓	✓
Ukrainian	Ukraine	✓	♦	✓	✓	✓	✓
Urdu	India			✓	✓		
	Pakistan	✓		✓	✓		
Uzbek (Cyrillic)	Uzbekistan	✓		✓	✓	✓	
Uzbek (Latin)	Uzbekistan	✓		✓	✓	✓	
Vietnamese	Viet Nam	✓	♦	✓	✓	✓	✓
Welsh	United Kingdom	✓		✓		✓	✓
Xhosa	South Africa			✓			
Zulu	South Africa			✓			✓