



**FREE eBook**

**LEARNING**

**internationalization**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#international  
alization**

# Table of Contents

<b>About</b> .....	<b>1</b>
<b>Chapter 1: Getting started with internationalization</b> .....	<b>2</b>
Remarks.....	2
Versions.....	2
Examples.....	2
Installation or Setup.....	3
<b>Chapter 2: i18n</b> .....	<b>5</b>
Examples.....	5
Language Neutral Date and Time Formatting in PHP via the MessageFormatter class.....	5
ICU4J Plurality for Scala via the MessageFormat class.....	5
<b>Credits</b> .....	<b>6</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [internationalization](#)

It is an unofficial and free internationalization ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official internationalization.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with internationalization

## Remarks

Internationalization is an engineering process intended to generalize a product so that it can handle multiple languages, scripts and cultural conventions without the need for redesign or recompilation. It involves a range of activities, such as:

- Prepare software distribution platforms for a global audience
- Defining mappings for currency conversion
- Defining sorting rules which change based on the current alphabet
- Defining numeric and date formatting which adapts to the current locale selected
- Discovering the language of the user's input, so that content can be displayed correctly
- Offering a default language, with the option to have the text transliterated from one language to another
- Permitting the user to select language-related user preferences
- Displaying times, dates, and calendar information in appropriate languages and formats

## Versions

Version	Release Date
ICU 58	2016-12-09
ICU 57	2016-03-23
ICU 56	2015-10-07
ICU 55	2015-04-01
ICU 54	2014-10-01
ICU 53	2014-04-02
ICU 52	2013-10-09
ICU 51	2013-03-22
ICU 50	2012-12-17
ICU 49	2012-03-21

## Examples

## Installation or Setup

ICU(International Components for Unicode) can be installed as a library for the following languages:

- [Java: ICU4J](#)
- [C/C++: ICU4C](#)
- [Python: pyICU](#)
- [Perl: Wrappers for ICU \(PICU\)](#)
- [C#: ICU-dotnet](#)
- [Erlang: icu4e](#)
- [Ruby: ffi-icu](#)
- [Haskell: text-icu](#)
- [Lua: ICU4Lua](#)

Here is a series of installation steps for a few implementations:

- ICU4J:
  - place icu4j.jar in your CLASSPATH
    - Optionally, place icu4j-charset.jar in your CLASSPATH for charset API support
  - To build ICU4J, you will need JDK 7+ and Apache Ant 1.9+
  - Set the JDK and Ant environment variables
    - set JAVA\_HOME=C:\jdk1.8.0
    - set ANT\_HOME=C:\apache-ant
    - set PATH=%JAVA\_HOME%\bin;%ANT\_HOME%\bin;%PATH%
  - Run the desired target defined in build.xml. The default target is "jar" which compiles ICU4J library class files and create ICU4J jar files. For example:
    - C:\icu4j>ant
- ICU4C
  - Place the path to ICU4C in the LD\_LIBRARY\_PATH environment variable so that dynamic linker can find it. For example:
    - export LD\_LIBRARY\_PATH=/opt/icu/lib
  - Or Windows:
    - set PATH=%PATH%;C:\icu\dist\bin; set INCLUDE=%INCLUDE%;C:\icu\dist\include; set LIB=%LIB%;C:\icu\dist\bin
  - Run the configure script and build ICU4C using a makefile:
    - ./configure --prefix=/opt/icu
  - Or CygWin:
    - set PATH=%PATH%;C:\Cygwin\bin C:> dos2unix \* C:> dos2unix -f configure
    - C:> "C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\vcvarsall.bat" x86
    - C:> bash runConfigureICU Cygwin/MSVC -prefix=/cygdrive/c/icu/dist
  - Run make and make install:
    - make && make install

[Read Getting started with internationalization online:](#)

<https://riptutorial.com/internationalization/topic/7109/getting-started-with-internationalization>

---

# Chapter 2: i18n

## Examples

### Language Neutral Date and Time Formatting in PHP via the MessageFormatter class

```
<?php

// Setting parameters
$time    = time();
$values = [7, $time, $time];

// Prints "At 3:50:31 PM on Apr 19, 2015, there was a disturbance on planet 7."
$pattern = "At {1, time} on {1, date}, there was a disturbance on planet {0, number}.";
$formatter = new MessageFormatter("en_US", $pattern);
echo $formatter->format($values);

// Prints "À 15:53:01 le 19 avr. 2015, il y avait une perturbation sur la planète 7."
$pattern = "À {1, time} le {1, date}, il y avait une perturbation sur la planète {0, number}.";
$formatter = new MessageFormatter("fr_FR", $pattern);
echo $formatter->format($values);

?>
```

### ICU4J Plurality for Scala via the MessageFormat class

```
import com.ibm.icu.text.MessageFormat
import java.util.Locale
import scala.collection.JavaConverters._

// Outputs "A common message for all options. Selected: (other) "
val formatted = new MessageFormat(
  "{messageCount, plural, one {{aMessage} (one)} other {{aMessage} (other)}}",
  new java.util.Locale("en_US")
).format(Map(
  "messageCount" -> 900,
  "aMessage" -> "A common message for all options. Selected: "
).asJava)
```

Read i18n online: <https://riptutorial.com/internationalization/topic/9859/i18n>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with internationalization	<a href="#">Community</a> , <a href="#">Paul Sweatte</a>
2	i18n	<a href="#">Paul Sweatte</a>