## How do I format date using a locale based format?

The code below demonstrate how to format date information for a specific Locale. In the example utilize the java.text.SimpleDateFormat class.

package org.kodejava.example.text;

02.

03.import java.util.Locale;

04.import java.util.Date;

05.import java.text.SimpleDateFormat;

06.

07.public class FormatDateLocale {

08.    public static void main(String[] args) {

09.        //

10.        // Defines an array of Locale we are going to use for

11.        // formatting date information.

12.        //

13.        Locale[] locales = new Locale[] {

14.                Locale.JAPAN,

15.                Locale.CHINA,

16.                Locale.KOREA,

17.                Locale.TAIWAN,

18.                Locale.ITALY,

19.                Locale.FRANCE,

20.                Locale.GERMAN

21.        };

22.

23.        // Get an instance of current date time

24.        Date today = new Date();

25.

26.        //

27.        // Iterates the entire Locale defined above and create a long

28.        // formatted date using the SimpleDateFormat.getDateInstance()

29.        // with the format, the Locale and the date information.

30.        //

31.        for (Locale locale : locales) {

32.            System.out.println("Date format in "

33.                + locale.getDisplayName()

34.                + " = "

35.                + SimpleDateFormat.getDateInstance(

36.                      SimpleDateFormat.LONG, locale)

37.                          .format(today).toUpperCase());

38.        }

39.    }

40.}

import java.applet.\*; //  
import java.text.\*; //  
  
/\*\* Illustrate use of the DecimalFormat class. \*\*/  
public class **DecimalFormatDemo** extends Applet  
{  
  
  String fValStr = "test";  
  public void init () {  
  
    double q = 1.0/3.0;  
  
    // Define the format pattern in a string  
    String fmt = "0.000";  
  
    // Create a DecimalFormat instance for this format  
    DecimalFormat df = new DecimalFormat (fmt);  
  
    // Create a string from the double according to the  
    // format  
    fValStr = df.format (q);  
  
    System.out.println ("1.0/3.0 = " + fValStr);  
  
    // Can change the format pattern:  
    df.applyPattern ("0.00000");  
    fValStr = df.format (q);  
    System.out.println ("1.0/3.0 = " + fValStr);  
  
    // The # symbol indicates trailing blanks  
    df.applyPattern ("0.######");  
    fValStr = df.format (1.0/2.0);  
    System.out.println ("1.0/2.0 = " + fValStr);  
  
  
    df.applyPattern ("0.00E0");  
    fValStr = df.format (1000.0/3.0);  
    System.out.println ("1000.0/3.0 = " + fValStr);  
  
  
    df.applyPattern ("0.00E0");  
    fValStr = df.format (3.0/4567.0);  
    System.out.println ("3.0/4567.0 = " + fValStr);  
  
  
    // Negative infinity  
    df.applyPattern ("0.000E0");  
    fValStr = df.format (-1.0/0.0);  
    System.out.println ("-1.0/0.0 = " + fValStr);  
  
    // NaN  
    df.applyPattern ("0.000E0");  
    fValStr = df.format (0.0/0.0);  
    System.out.println ("0.0/0.0 = " + fValStr);  
  
  }  
  
  
  // Paint message in Applet window.  
  public void paint (java.awt.Graphics g) {  
  
     g.drawString (fValStr,20,20);  
  
  }  
  
  // Can also run this program as an application.  
  // No windowing needed so just run the applets  
  // init () function.  
  public static void main (String [] args) {  
    DecimalFormatDemo obj = new DecimalFormatDemo ();  
    obj.init ();  
  }  
  
} // class DecimalFormatDemo

/\*

Output of this program:

1.0/3.0 = 0.333  
1.0/3.0 = 0.33333  
1.0/2.0 = 0.5  
1000.0/3.0 = 3.33E2  
3.0/4567.0 = 6.57E-4  
-1.0/0.0 = -?  
0.0/0.0 = ?

\*/

**import**java.text.\*;  
**import**java.util.\*;  
  
**class**NumberFormatExample {  
  
    **public void**numberFormat(Locale currentLocale) {  
        Integer intNum = **new**Integer(123456789);  
        Double doubleNum = **new**Double(1234.1234);  
  
        String intNumOut, doubleNumOut;  
  
        NumberFormat numberFormatter =   
        NumberFormat.getNumberInstance(currentLocale);

        intNumOut = numberFormatter.format(intNum);  
        doubleNumOut = numberFormatter.format(doubleNum);  
  
        System.out.println();  
        System.out.println("Integer num : " + intNumOut +   
        "   " + currentLocale.toString());

        System.out.println("Double  num : " + doubleNumOut +   
        "   " + currentLocale.toString());

    }  
  
    **public static void**main(String args[]) {  
        Locale[] locales = **new**Locale[]{**new**Locale  
        ("fr", "FR"), **new**Locale("de", "DE"),

**new**Locale("ca", "CA"),new Locale("rs", "RS"),  
            new Locale("en", "IN")  
        };  
  
        NumberFormatExample[] formate = **new** NumberFormatExample[locales.length];

**for**(**int**i = 0; i < locales.length; i++) {  
            formate[i] = **new**NumberFormatExample();  
            formate[i].numberFormat(locales[i]);  
  
        }  
    }  
}

import java.text.\*;  
  
public class FormattingUsingCustomFormat {  
        public static void main(String[] args) {  
                NumberFormat nf1 = new DecimalFormat("000000");  
                String st1 = nf1.format(-1234.567);  
                System.out.println(st1); //-001235  
                NumberFormat nf2 = new DecimalFormat("##");  
                String st2 = nf2.format(-1234.567);  
                System.out.println(st2); //-1235  
                NumberFormat nf3 = new DecimalFormat("##00");  
                String st3 = nf3.format(0);  
                System.out.println(st3); //00  
                NumberFormat nf4 = new DecimalFormat(".00");  
                String st4 = nf4.format(-.567);  
                System.out.println(st4); //-.57  
                NumberFormat nf5 = new DecimalFormat("0.00");  
                String st5 = nf5.format(-.567);  
                System.out.println(st5); //-0.57  
                NumberFormat nf6 = new DecimalFormat("#.#");  
                String st6 = nf6.format(-1234.567);  
                System.out.println(st6); //-1234.6  
                NumberFormat nf7 = new DecimalFormat("#.####");  
                String st7 = nf7.format(-1234.567);  
                System.out.println(st7); //-1234.567  
                NumberFormat nf8 = new DecimalFormat(".##");  
                String st8 = nf8.format(-1234.567);  
                System.out.println(st8); //-1234.57  
                NumberFormat nf9 = new DecimalFormat("#.000000");  
                String st9 = nf9.format(-1234.567);  
                System.out.println(st9); // -1235  
                NumberFormat nf10 = new DecimalFormat("#,###,###");  
                String st10 = nf10.format(-1234.567);  
                System.out.println(st10); -1,234.567000  
                NumberFormat nf11 = new DecimalFormat("#;(#)");  
                String st11 = nf11.format(-1234.567);  
                System.out.println(st11); // (1235)  
                NumberFormat nf12 = new DecimalFormat("'#'#");  
                String st12 = nf12.format(-1234.567);  
                System.out.println(st12); // -#1235  
                NumberFormat nf13 = new DecimalFormat("'abc'#");  
                String st13 = nf13.format(-1234.567);  
                System.out.println(st13); // -abc1235  
        }  
}

/\* -001235  
-1235  
00  
-.57  
-0.57  
-1234.6  
-1234.567  
-1234.57  
-1234.567000  
-1,235  
(1235)  
-#1235  
-abc1235

\*/

Formatter: format('% d', -100) (the space format specifiers)

/\*  
-100  
 100  
-200  
 200  
  
 \*/  
**import**java.util.Formatter;  
  
**public class**MainClass {  
  **public static void**main(String args[]) {  
    Formatter fmt = **new**Formatter();  
    fmt.format("% d", -100);   
    System.out.println(fmt);   
   
    fmt = **new**Formatter();   
    fmt.format("% d", 100);   
    System.out.println(fmt);   
   
    fmt = **new**Formatter();   
    fmt.format("% d", -200);   
    System.out.println(fmt);   
   
    fmt = **new**Formatter();   
    fmt.format("% d", 200);   
    System.out.println(fmt);   
  }  
}