**Date Formatting In Java With Time Zone**

[**JAVA DATE**](http://beginnersbook.com/category/technology/java-guide/java-date/)

This tutorial will help you getting the [**current time**](http://beginnersbook.com/2013/05/current-date-time-in-java/), date and day in any given format for any particular time zone in java.  
Listed below are some IDs for some common **Time zones** in the US:

|  |  |
| --- | --- |
| **Time Zone** | **Java Time Zone ID** |
| Hawaiian Standard Time | US/Hawaii |
| Alaska Standard Time | US/Alaska |
| Pacific Standard Time | US/Pacific |
| Mountain Standard Time | US/Mountain |
| Central Standard Time | US/Central |
| Eastern Standard Time | US/Eastern |

Java 6 makes available 575 different time zone IDs for places around the world.  It is recommended that Java Time zone IDs mentioned in the above table are used, instead of using three letter time zone abbreviations like “EST”, “PDT”, etc.  as they are not unique in their usage around the world.

For Example, CST can stand for both Central Standard Time and China Standard Time.

Though Java can still recognize these for compatibility with Java 1.1, Usage of these abbreviations in JREs may produce unpredictable results. Currently Java uses the “tz database” for listing of time zone related to a particular location.

**Sample Code for obtaining the list of Time Zones in the US:**

For the purpose of learning, “US/Eastern” has been used in the code in the below example.The functions used in the **below complete code**, takes in the values of the **date format and the time zone**. The user needs to decide the format in which the date/day/time value is required and the target time zone. For date formatting you can refer [**Date formatting in java**](http://beginnersbook.com/2013/04/java-date-format/).

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.TimeZone;

public class ObtainDate

{

public static void main(String[] args)

{

ObtainDate obtainDate = new ObtainDate();

TimeZone timeZone = TimeZone.getTimeZone("US/Eastern");

String dateFormat = "MMMM dd,yyyy G"; //MMMM dd,yyyy G

String timeFormat = "hh:mm:ss.SSS a zzzz";

String dayFormat = "EEEEEE";

System.out.println("Todays Day:" + obtainDate.getTodaysDay(dayFormat,timeZone));

System.out.println("Todays Date:" + obtainDate.getTodayDate(dateFormat,timeZone));

System.out.println("Current Time:" + obtainDate.getCurrentTime(timeFormat,timeZone));

}

/\*\*

\* Description - Method to Get Today's day

\* @author Chaitanya

\* @param dateFormat

\* @param TimeZone

\*/

public String getTodaysDay(String dayFormat, TimeZone timeZone)

{

Date date = new Date();

/\* Specifying the format \*/

DateFormat requiredFormat = new SimpleDateFormat(dayFormat);

/\* Setting the Timezone \*/

requiredFormat.setTimeZone(timeZone);

/\* Picking the day value in the required Format \*/

String strCurrentDay = requiredFormat.format(date).toUpperCase();

return strCurrentDay;

}

/\*\*

\* Description - Method to Get Current time

\* @author Chaitanya

\* @param dateFormat

\* @param TimeZone

\*/

public String getCurrentTime(String timeFormat, TimeZone timeZone)

{

/\* Specifying the format \*/

DateFormat dateFormat = new SimpleDateFormat(timeFormat);

/\* Setting the Timezone \*/

Calendar cal = Calendar.getInstance(timeZone);

dateFormat.setTimeZone(cal.getTimeZone());

/\* Picking the time value in the required Format \*/

String currentTime = dateFormat.format(cal.getTime());

return currentTime;

}

/\*\*

\* Description - Method to Get Today's date

\* @author Chaitanya

\* @param dateFormat

\* @param TimeZone

\*/

public String getTodayDate(String dateFormat, TimeZone timeZone)

{

Date todayDate = new Date();

/\* Specifying the format \*/

DateFormat todayDateFormat = new SimpleDateFormat(dateFormat);

/\* Setting the Timezone \*/

todayDateFormat.setTimeZone(timeZone);

/\* Picking the date value in the required Format \*/

String strTodayDate = todayDateFormat.format(todayDate);

return strTodayDate;

}

}

Output:

Todays Day:MONDAY

Todays Date:January 06,2014 AD

Current Time:11:34:22.173 AM Eastern Standard Time

**Code Skeleton:**  
The basic steps in the calculation of day/date/time in a particular time zone:

* Obtain a new Date object
* A date format object is created in Simple Date Format
* The required format is assigned to the date format object – For Eg: hh:mm:ss.SSS
* Set the required time zone to the date format object
* Pick the day/date/time value from the date format object by passing the date object created in the first step.

The table below gives the various date patterns and their significance in Java Timestamp Format:

|  |  |
| --- | --- |
| **Pattern** | **Denotes** |
| yyyy | Current Year |
| MM | Current Month in number |
| MMM | Current Month abbreviated |
| MMMM | Current Month in Full |
| dd | Current Date of the month |
| DD | Current Day of the year |
| hh | Current Time’s Hour (12 Hour clock) |
| HH | Current Time’s Hour (24 Hour clock) |
| a | AM/PM |
| mm | Current Time’s Minute |
| ss | Current Time’s Second |
| SSS | Current Time’s Millisecond (the number of SSS can be incremented to obtain the fraction of the second) |
| G | Epoch |
| zzz | Time Zone abbreviated |
| zzzz | Time Zone in Full |
| EEEEEE | Current Day in Full |
| EEE | Current Day abbreviated |

A reference like **yyyy.MM.dd G hh:mm:ss**would yield an output like**2011.11.07 AD 1:11:57 PM**. At the same time yyyy.MMM.DD  G HH:mm.SSS would yield output of **2011.JUL.194 AD 23:11.938**.