

```
/* -----  
/*  
/* File created on 23/09/2016  
/* Version: 1.00  
/* For any question about the content please e-mail to <TAXUD.e-CUSTOMS.Technical-  
Specs@ec.europa.eu>.  
/*  
/* Published on CIRCABC (https://circabc.europa.eu/w/browse/00205ce8-8219-4643-9c6c-ab108e0bb836)  
/* CIRCABC > Taxation & Customs Union > e-Customs - IT aspects || Library > 02_NCTS >  
03_TECHNICAL_SPECS > 01_DDNTA  
/*  
/* -----  
/*  
/* This file will be part of the DDNTA v20.00 and following version. It's related to the RFC-  
List.29 #392  
/* It includes the code received by DG TAXUD on 07.04.2016 from IRU, in order to validate  
the TIR Carnet Number.  
/* The MS must apply the same algorithm to validate the message received from traders.  
/* This code was provided free of charge and without commitment of IRU.  
/* Note: IRU also provides web services that can be used to obtain all information available  
for a dpecific TIR Guarantee.  
/*  
/* -----
```

```
////////////////////////////////////
```

```
package org.iru.tir.utils;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import java.util.Locale;
```

```
import java.util.regex.Matcher;
```

```
import java.util.regex.Pattern;
```

```

public class TIRCarnetUtils {

    private static final long LAST_TIR_CARNET_NUMBER_WITHOUT_CONTROL_CHARS =
25000000L;

    private static final Pattern TIR_CARNET_NUMBER_PATTERN = Pattern.compile("^[A-
Z]{2}?(\\d+)$");

    public final static String getCheckedTIRCarnetNumber(long tirCarnetNumber) {

        if (tirCarnetNumber <= LAST_TIR_CARNET_NUMBER_WITHOUT_CONTROL_CHARS)

            return Long.toString(tirCarnetNumber);

        long carnetModulo23 = tirCarnetNumber % 23;

        int checkCharacter = ((int) 'A') + new Long((3 * carnetModulo23 + 17) % 26).intValue();

        if (carnetModulo23 < 12) {

            return String.valueOf(Character.toChars(checkCharacter)) + "X" +
Long.toString(tirCarnetNumber);

        }

        return "X" + String.valueOf(Character.toChars(checkCharacter)) +
Long.toString(tirCarnetNumber);

    }

    public final static Long getNumericTIRCarnetNumber(String checkedTirCarnetNumber)
throws NumberFormatException {

        if (checkedTirCarnetNumber == null || "".equals(checkedTirCarnetNumber)) {

            return null;

        }

        String tirCarnetNumber =
checkedTirCarnetNumber.toUpperCase(Locale.ENGLISH).replaceAll("/\\s/g", "");

        Matcher m = TIR_CARNET_NUMBER_PATTERN.matcher(tirCarnetNumber);

        if (m.matches()) {

            String cd = m.group(1);

```

```
        long num = Long.parseLong(m.group(2));
        if (num > 25000000L) {
            if (cd != null && cd.equals(getCheckedTIRCarnetNumber(num).substring(0, 2)))
                return num;
        } else {
            if (cd == null)
                return num;
        }
    }

    throw new NumberFormatException("Wrong Checked TIR Carnet Number: " +
checkedTirCarnetNumber);
}
```

```
public final static List<Integer> getNumericTIRCarnetNumber(List<String>
checkedTirCarnetNumbers) {
    List<Integer> result = new ArrayList<Integer>(checkedTirCarnetNumbers.size());
    for (String cn : checkedTirCarnetNumbers) {
        result.add(getNumericTIRCarnetNumber(cn).intValue());
    }
    return result;
}
```

```
////////////////////////////////////
```

```
/* -----
```