

# Why is January 1 being reported as the last week of the previous year?

---

 [devblogs.microsoft.com/oldnewthing/20151112-00](http://devblogs.microsoft.com/oldnewthing/20151112-00)

November 12, 2015



Raymond Chen

A customer (via the customer liaison) reported a problem that occurred when their program was run in Germany.

Ah, those pesky Germans. Always causing trouble.

In Germany, we find that the week numbers are wrong. For example, for the date January 1, 2005, the `GetLocaleInfo` function reports `FIRSTDAYOFWEEK` as 53 instead of the expected value of 1. The attached program demonstrates the issue.

The customer wants to know if there is anything we can do so we get the correct result. We saw the `SetLocaleInfo` function but weren't sure if it can be used to get the English calendar on a German system, or what unintended consequences there may be.

```

using System;
using System.Globalization;

class Program
{
    static void RunScenario()
    {
        DateTimeFormatInfo dfi = DateTimeFormatInfo.CurrentInfo;
        Calendar cal = dfi.Calendar;

        DateTime date = new DateTime(2005, 1, 1);
        int weekOfYear = cal.GetWeekOfYear(date, dfi.CalendarWeekRule,
                                           dfi.FirstDayOfWeek);
        Console.WriteLine("{0:d} is week number {1}", date, weekOfYear);
    }

    static void Main()
    {
        RunScenario();

        System.Threading.Thread.
        CurrentThread.CurrentCulture = new CultureInfo("da-DK");
        RunScenario();
    }
}

```

One thing I noticed was that it wasn't the Germans causing trouble at all. It's the Danes!

```
... new CultureInfo("da-DK");
```

So your frustration is misdirected. You should be upset at those pesky Danes.

But the first thing I noticed is that the question makes no sense. The `FIRSTDAYOFWEEK` is an integer in the range 0 through 6, where 0 means Monday, and 6 means Sunday. It never returns 53. What's more, the `GetLocaleInfo` function is never called by this program.

So let's ignore the stated question and try to infer the question from the code. The code prints the week number for January 1, 2005, according to US-English and Danish rules. For US-English, it returns week 1, and for Denmark, it returns week 53.

This is hardly surprising, because if you play around in the Immediate window, or just add some more `WriteLine` statements, you'll see that the US-English and Danish locales have different calendar preferences:

Property	US-English	Danish
FirstDayOfWeek	Sunday	Monday

CalendarWeekRule	FirstDay	FirstFourDayWeek
------------------	----------	------------------

How did I know to look at those properties? Because those are the properties that the program itself passes to the `GetWeekOfYear` function!

Denmark starts the week on Monday. It follows the rule that Week 1 of the year is the first week with at least four days from that year. And January 1, 2005 was a Saturday. This means that the week containing January 1, 2005 has only two days from 2005. That's not enough, so the the first week of the year begins on January 3, 2005. January 1, 2005 is considered to be part of the last week of 2004, which is week number 53.

Therefore, the value of 53 is correct. If you want Denmark to follow week-numbering rules consistent with the United States, you will need to take it up with [the Folketing](#), and possibly [the European Parliament](#).

Now, if you decide that you want to show week numbers according to the United States convention even to people who did nothing wrong aside from happening to live in a country with the wrong week numbering rules, you can just pass your custom week numbering rules to the `GetWeekOfYear` function.

```
int weekOfYear = cal.GetWeekOfYear(date,
    CalendarWeekRule.FirstDay,
    DayOfWeek.Sunday);
```

Don't change the system locale; that is trying to apply a global solution to a local problem.

Mind you, your local solution is going to make everybody in Denmark say, "How come your program gets all the week numbers wrong?"

**Bonus reading:** [ISO 8601 Week of Year Format in Microsoft .NET](#).

[Raymond Chen](#)

**Follow**

