

第54章 CultureInfoクラス



CultureInfoクラスは、言語、国、地域、暦、慣習などを表します。

前章までと関連深いのは暦です。

DateTime構造体のToStringメソッドの4つのオーバーロードバージョンのうち、2つはすでに前章までにやりました。残り2つは次のようなものです。

```
public string ToString (
    string format,
    IFormatProvider provider
)
public string ToString (
    IFormatProvider provider
)
```

formatには、書式指定文字列を指定します。

providerには、CultureInfoクラスのインスタンスを指定します。

System.GlobalizationクラスはSystem.Globalization名前空間で定義されています。

CultureInfoクラスには、4つのオーバーロードされたコンストラクタがあります。

```
public CultureInfo (
    int culture
)
public CultureInfo (
    string name
)
public CultureInfo (
    int culture,
    bool useUserOverride
)
public CultureInfo (
    string name,
    bool useUserOverride
)
```

cultureには、カルチャ識別子を指定します。

nameには、カルチャ名を指定します。

useUserOverrideは、trueでユーザーが選択したカルチャ設定、falseで規定のカルチャ設定となります。カルチャ名は、たとえば日本語-日本なら"ja-JP"、英語-米国なら"en-US"、英語-英国なら"en-GB"となります。これらについて、それぞれ識別子がついています。ja-Jpは0x0411、en-USは0x0409、en-GBは0x0809といった具合です。必要に応じてMSDNのライブラリ等で調べてみてください。

では、簡単なサンプルを見てみましょう。

```
// culture01.cs
```

```
using System;
using System.Globalization;

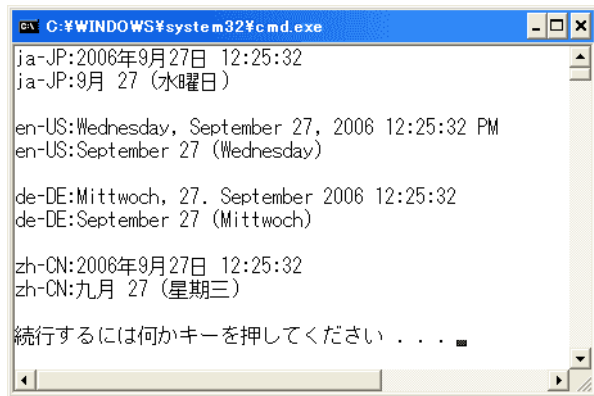
class culture01
{
    public static void Main()
    {
        CultureInfo ci;
        DateTime now = System.DateTime.Now;

        string[] strcul = {
            "ja-JP", "en-US", "de-DE", "zh-CN" };

        for (int i = 0; i < strcul.Length; i++)
        {
            ci = new CultureInfo(strcul[i]);
            Console.WriteLine("{0}:{1}", strcul[i], now.ToString("F", ci));
            Console.WriteLine("{0}:{1}", strcul[i], now.ToString("MMM dd (dddd)", ci));
            Console.WriteLine("Yn");
        }
    }
}
```

de-DEはドイツ語-ドイツ、zh-CNは中国語-中国です。

実行結果は次のようになります。



日本語表示の場合、西暦ではなく元号で表示したい場合も多いと思われます。

CultureInfoクラスには、DateTimeFormatプロパティがあります。このプロパティは このクラスのDateTimeFormatInfo型の値を取得設定します。

DateTimeFormatInfoクラスには、Calendarプロパティを持っており、カルチャで使用する 暦の取得、設定を行います。

このCalendarプロパティに日本語の暦を設定すれば、元号表示ができます。うまい具合に JapaneseCalendarクラスというのがあります。このインスタンスを先ほどのCalendarプロパティに設定すればよいことになります。

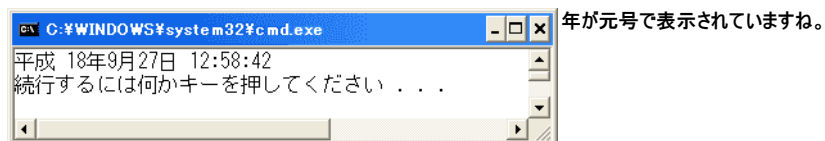
```
CultureInfo ci = new CultureInfo("ja-JP");  
ci.DateTimeFormat.Calendar = new JapaneseCalendar();
```

と、このような感じになります。

では、サンプルを見てみましょう。

```
// calendar01.cs  
  
using System;  
using System.Globalization;  
  
class calendar01  
{  
    public static void Main()  
    {  
        CultureInfo ci = new CultureInfo("ja-JP");  
        DateTime dt = DateTime.Now;  
  
        ci.DateTimeFormat.Calendar = new JapaneseCalendar();  
  
        Console.WriteLine(dt.ToString("F", ci));  
    }  
}
```

実行結果は次のようになります。



年が元号で表示されていますね。