

Thread.CurrentUICulture Property

Namespace: [System.Threading](#)

Assemblies: mscorlib.dll, netstandard.dll, System.Threading.Thread.dll

In this article

[Definition](#)

[Examples](#)

[Remarks](#)

[Applies to](#)

Gets or sets the current culture used by the Resource Manager to look up culture-specific resources at run time.

C#

 Copy

```
public System.Globalization.CultureInfo CurrentUICulture { get; set; }
```

Property Value

[CultureInfo](#)

An object that represents the current culture.

Exceptions

[ArgumentNullException](#)

The property is set to `null`.

[ArgumentException](#)

The property is set to a culture name that cannot be used to locate a resource file. Resource filenames must include only letters, numbers, hyphens or underscores.

InvalidOperationException

.NET Core only: Reading or writing the culture of a thread from another thread is not supported.

Examples

The following example determines whether the language of the current thread's UI culture is French. If it is not, it sets the UI culture of the current thread to English (United States).

C#

 Copy

```
using System;
using System.Globalization;
using System.Threading;

public class Example
{
    public static void Main()
    {
        // Change the current culture if the language is not French.
        CultureInfo current = Thread.CurrentThread.CurrentUICulture;
        if (current.TwoLetterISOLanguageName != "fr") {
            CultureInfo newCulture = CultureInfo.CreateSpecificCulture("en-US");
            Thread.CurrentThread.CurrentUICulture = newCulture;
            // Make current UI culture consistent with current culture.
            Thread.CurrentThread.CurrentCulture = newCulture;
        }
        Console.WriteLine("The current UI culture is {0} [{1}]",
            Thread.CurrentThread.CurrentUICulture.NativeName,
            Thread.CurrentThread.CurrentUICulture.Name);
        Console.WriteLine("The current culture is {0} [{1}]",
            Thread.CurrentThread.CurrentUICulture.NativeName,
```

```
        Thread.CurrentThread.CurrentCulture.Name);  
    }  
}  
// The example displays the following output:  
//     The current UI culture is English (United States) [en-US]  
//     The current culture is English (United States) [en-US]
```

The following code example shows the threading statement that allows the user interface of a Windows Forms to display in the culture that is set in Control Panel. Additional code is needed.

C#

 Copy

```
using System;  
using System.Threading;  
using System.Windows.Forms;  
  
class UICulture : Form  
{  
    public UICulture()  
    {  
        // Set the user interface to display in the  
        // same culture as that set in Control Panel.  
        Thread.CurrentThread.CurrentCulture =  
            Thread.CurrentThread.CurrentCulture;  
  
        // Add additional code.  
    }  
  
    static void Main()  
    {  
        Application.Run(new UICulture());  
    }  
}
```

Remarks

The UI culture specifies the resources an application needs to support user input and output, and by default is the same as the operating system culture. See the [CultureInfo](#) class to learn about culture names and identifiers, the differences between invariant, neutral, and specific cultures, and the way culture information affects threads and application domains. See the [CultureInfo.CurrentCulture](#) property to learn how a thread's default UI culture is determined.

❗ Important

The [CurrentUICulture](#) property doesn't work reliably when used with any thread other than the current thread. In .NET Framework, reading the property is reliable, although setting it for a thread other than the current thread is not. On .NET Core, an [InvalidOperationException](#) is thrown if a thread attempts to read or write the [CurrentUICulture](#) property on a different thread. We recommend that you use the [CultureInfo.CurrentCulture](#) property to retrieve and set the current culture.

The [CultureInfo](#) returned by this property can be a neutral culture. Neutral cultures should not be used with formatting methods such as [String.Format\(IFormatProvider, String, Object\[\]\)](#), [DateTime.ToString\(String, IFormatProvider\)](#), and [Convert.ToString\(Char, IFormatProvider\)](#). Use the [CultureInfo.CreateSpecificCulture](#) method to get a specific culture, or use the [CurrentCulture](#) property.

ⓘ Note

The [CultureInfo.CreateSpecificCulture](#) method throws [ArgumentException](#) for the neutral cultures "zh-Hant" ("zh-CHT") and "zh-Hans" ("zh-CHS").

Applies to

.NET Core

3.1, 3.0, 2.2, 2.1, 2.0

.NET Framework

4.8, 4.7.2, 4.7.1, 4.7, 4.6.2, 4.6.1, 4.6, 4.5.2, 4.5.1, 4.5, 4.0, 3.5, 3.0, 2.0, 1.1

.NET Standard

2.1, 2.0

Xamarin.Android

7.1

Xamarin.iOS

10.8

Xamarin.Mac

3.0

Is this page helpful?

 Yes  No
