

The Turkish lira's currency code is an unexpected source of problems with computer programmers

devblogs.microsoft.com/oldnewthing/20190912-00

September 12, 2019



Raymond Chen

The modern Turkish lira was introduced in 2005, replacing the old Turkish lira at an exchange rate of one million old lira to one new lira.

The ISO-4217 currency code for the old lira was TRL, following the currency code naming convention of taking the first two letters of the currency code from the ISO 3166-1 alpha-2 country code (TR = Turkey) and the final letter of the currency code from the name of the currency (L = lira).¹

The ISO-4217 currency code for the new lira follows the same pattern. The first two letters are TR, representing Turkey, and the final letter comes from the official name of the new currency during the transition period: *Yeni Türk lirası* (*yeni* = new), leading to the currency code TRY.

I think you see where this is going.

It's not uncommon to see a macro called `TRY`, usually related to exception handling.

In the presence of such a macro, weird and bizarre things tend to happen if you include a header file that contains definitions for currency codes, or if you yourself are trying to define a class that deals with currency codes.

```
enum class Currency
{
    AED,
    AFN,
    ...
    TRY, // incomprehensible compiler error here
    ...
};
```

One example of this collision is between MFC's TRY/CATCH macros and the Windows.Globalization.CurrencyIdentifiers.TRY property.

One way out of this problem is to get rid of your `TRY` macro, or at least rename it.

If you really need that `TRY` macro, you can try including the header files in the opposite order, so that the code that tries to use an identifier named `TRY` will compile successfully, and then later you swoop in and change the definition. It means that the `TRY` identifier will be inaccessible, but presumably you weren't using it, or you wouldn't have defined a macro called `TRY` in the first place.

Yet another option is to temporarily undefine the `TRY` macro.

```
#pragma push_macro("TRY")
#undef TRY
#include <winrt/Windows.Globalization.h>
#pragma pop_macro("TRY")
```

The `push_macro` and `pop_macro` pragmas are nonstandard, but they are supported by Visual C++, gcc, and clang.

¹ If the result matches an existing currency code, then hunt around for a different final letter. This hunt could be quite extensive for countries that have undergone many changes of official currency.

Raymond Chen

Follow

