Converting between LCIDs and RFC 1766 language codes

devblogs.microsoft.com/oldnewthing/20060105-00

January 5, 2006

Raymond Chen

Occasionally, I see someone ask for a function that converts between LCIDs (such as 0x0409 for English-US) and RFC 1766 language identifiers (such as “en-us”). The rule of thumb is, if it’s something a web browser would need, and it has to do with locales and languages, you should look in the MLang library. In this case, the IMultiLanguage::GetRfc1766FromLcid method does the trick.

For illustration, here’s a program that takes US-English and converts it to RFC 1766 format. For fun, we also convert “sv-fi” (Finland-Swedish) to an LCID.

```
#include <stdio.h>
#include <ole2.h>
#include <oleauto.h>
#include <mlang.h>
int __cdecl main(int argc, char **argv)
{
 HRESULT hr = CoInitialize(NULL);
 if (SUCCEEDED(hr)) {
  IMultiLanguage * pml;
  hr = CoCreateInstance(CLSID_CMultiLanguage, NULL,
                        CLSCTX_ALL,
                        IID_IMultiLanguage, (void**)&pml);
  if (SUCCEEDED(hr)) {
   // Let's convert US-English to an RFC 1766 string
   BSTR bs;
   LCID lcid = MAKELCID(MAKELANGID(LANG_ENGLISH,
                        SUBLANG_ENGLISH_US), SORT_DEFAULT);
   hr = pml->GetRfc1766FromLcid(lcid, &bs);
   if (SUCCEEDED(hr)) {
    printf("%ws\n", bs);
    SysFreeString(bs);
   }
   // And a sample reverse conversion just for good measure
   bs = SysAllocString(L"sv-fi");
   if (bs && SUCCEEDED(pml->GetLcidFromRfc1766(&lcid, bs))) {
    printf("%x\n", lcid);
   }
   SysFreeString(bs);
   pml->Release();
  }
  CoUninitialize();
 }
 return 0;
}
```

When you run this program, you should get

```
en-us
81d
```

“en-us” is the RFC 1766 way of saying “US-English”, and 0x081d is `MAKELCID(MAKELANGID(LANG_SWEDISH, SUBLANG_SWEDISH_FINLAND), SORT_DEFAULT)`.

If you browse around, you’ll find lots of other interesting functions in the MLang library. You may recall that earlier we saw how to use MLang to display strings without those ugly boxes.

Update (January 2008): The globalization folks have told me that they’d prefer that people didn’t use MLang. They recommend instead the functions `LCIDToLocaleName` and `LocaleNameToLCID`. The functions are built into Windows Vista and are also available downlevel via a redistributable.

Raymond Chen

Follow