

Portable code, .Net and C#

While working on my FreeFrame wrapper I encountered an interesting situation. The only function exported by the dll of a FreeFrame video effect has a signature that only uses "unsigned int" in order to ease portability. But this function is used to return (single precision) floating-point values (stored in an unsigned int). Unsigned integers in the .Net world are not CLS-compliant so I created a COM interface in which these functions return float values and I just make the translation in the COM component using the familiar C/C++ code:

```
float * fptr = (float *)&tmpResult; //tmpResult is unsigned int
```

```
*finalResult = *fptr;
```

Then I realized that I could use the same code with C# in an unsafe method. I tried and I checked that the assembly code for the C/C++ version and the C# version is identical (a "lea" instruction followed by a "mov"). Decidedly, we need C/C++ in fewer and fewer places.