
Subject: Re: IsNullInstance() for Callback
Posted by [hojtsy](#) on Sat, 11 Mar 2006 19:54:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

I see the analogy. But I would write that code differently. `typedef void (*VoidFunctionPointer)();`
`VoidFunctionPointer operationPointer = NULL;`

.....
`if(NULL == operationPointer) {`
} This version is longer and but readers have a chance of understanding what the `if()` does without seeing the definition of the pointer variable, which could be pages away or in a different file. With this descriptive coding style code comments could be shorter, but yes I would put comments around any function pointer usage if the code is to be read by anyone else than myself. This area is not that well understood by the average programmers.
Now let's see how can I turn back this analogy to Callback. The best version I could come up with is: `Callback operationCallback;`

.....
`if((bool) operationCallback) {`
I know that `(bool)` is unnecessary, but implicit conversions just make the situation less understandable for average programmers. With seeing just the `if` line some of them will start to think that `operationCallback` is `bool`, or maybe `int`. This way I can convey the info that `operationCallback` is not `bool`, and they should go and check it's type. But it is still quite cryptic. The best notation would be something which could be read aloud as it's own explanation. Such notation is:

`if(operationCallback.IsValid()) {`
Now try to read out aloud this one, and then the one without `IsValid`.
Adding this one method will only have minimal effect on the readability of a big application, so it's not that important. I just wanted to explain my point of view.
