

Let's say we start out with two classes:

```
class IDevice {
public:
    virtual char const *GetName(void) = 0;
    virtual unsigned GetType(void) = 0;
};

class IKeyboard : public IDevice {
public:
    virtual bool IsCapsLockOn(void) = 0;
};
```

An object of type **IDevice** has a pointer at the beginning of it that points to the **IDevice-vtable**:

top_offset	Offset = -2
address of type_info	Offset = -1
address of GetName	Offset = 0
address of GetType	Offset = 1

An object of type **IKeyboard** has a pointer at the beginning of it that points to the **IKeyboard-vtable**:

top_offset	Offset = -2
address of type_info	Offset = -1
address of GetName	Offset = 0
address of GetType	Offset = 1
address of IsCapsLockOn	Offset = 2

The following code snippet:

```
int main(void)
{
    bool (IKeyboard::*p)(void) = &IKeyboard::IsCapsLockOn;
}
```

creates a member function pointer with the 'vtable offset' set to 2, because **IsCapsLockOn** is at offset 2.

## SEE NEXT PAGE

If we were to add a new method to 'IDevice' as follows:

```
class IDevice {
public:
    virtual char const *GetName(void) = 0;
    virtual unsigned GetType(void) = 0;
    virtual bool IsEnabled(void) = 0; // new method
};

class IKeyboard : public IDevice {
public:
    virtual bool IsCapsLockOn(void) = 0;
};
```

Then the two vtables would change as follows:

**IDevice-vtable:**

top_offset	Offset = -2
address of type_info	Offset = -1
address of GetName	Offset = 0
address of GetType	Offset = 1
address of IsEnabled	Offset = 2

**IKeyboard-vtable:**

top_offset	Offset = -2
address of type_info	Offset = -1
address of GetName	Offset = 0
address of GetType	Offset = 1
address of IsEnabled	Offset = 2
address of IsCapsLockOn	Offset = 3

We have an ABI break here because the 'vtable offset' of **IsCapsLockOn** has changed from **2** to **3**.

**SEE NEXT PAGE**

So we mark 'IDevice' as **extensible** and then mark the point at which it is extended:

```
class IDevice extensible {
public:
    virtual char const *GetName(void) = 0;
    virtual unsigned GetType(void) = 0;
    ++extensible++
    virtual bool IsEnabled(void) = 0;
};

class IKeyboard : public IDevice {
public:
    virtual bool IsCapsLockOn(void) = 0;
};
```

The vtables are laid out differently now. The vtable for IDevice now has the address of the *'extended method table'* prepended to it.

address of extended method table	Offset = -3
top_offset	Offset = -2
address of type_info	Offset = -1
address of GetName	Offset = 0
address of GetType	Offset = 1

And here's the *'extended method table'* for IDevice:

**IDevice-extended-methods-table:**

address of IsEnabled	Offset = 0
----------------------	------------

In order to accommodate this new layout, *member function pointers* will become a little more complicated. Previously a *member function pointer* only had a *'vtable offset'*, but now they will also have one bit to indicate whether the method's address resides as normal in the vtable, or whether it is to be found in the *'extend method table'*.

The class IKeyboard inherits from IDevice, and so IKeyboard will also need to have an *'extended method table'* prepended to its vtable.