

There are two additional options you missed out:

3- Catch unhandled key presses in the Ctrl's owner/parent. ie:

```
virtual void Key(dword key, int count)
```

```
{  
    if (key == K_RETURN && password_field.HasFocus())  
        login.PseudoPush();  
    else  
        return false;  
    return true;  
}
```

} Since you'd have to declare a new function to handle a callback anyway this isn't really any extra code.

4- Create a simple template to automate callback functionality:

```
template <class T>  
struct WithCallbacks : public T {  
    Callback WhenLeftDown  
  
    virtual void LeftDown(Point p, dword keyflags) { WhenLeftDown(); }  
}  
...
```

```
WithCallbacks<ArrayCtrl> array;
```

Given all the possible methods it doesn't seem worth retro-actively adding callbacks everywhere.

OTOH, the K\_RETURN event for EditField is common enough to warrant it's addition. It should be called WhenReturn or maybe WhenKeyReturn to match the key enum though. WhenEnter is too easy to confuse with the focus changing IMO.