Subject: Re: Having my HWND and eating it, too...
Posted by tvanriper on Thu, 11 Oct 2007 17:07:41 GMT
View Forum Message <> Reply to Message

Well, some example code for the trick I used...

Here's the child that creates the HWND...

```
class ChildCtrl : public Upp::Ctrl
{
public:
 ChildCtrl() : Upp::Ctrl() {};
 void Initialize( HWND owner, const Upp::Ctrl& ctrl )
 {
  SetRect( ctrl.GetRect() );
  Create( owner, WS_CHILD | WS_VISIBLE, 0, false, SW_SHOW, true );
 };
};
```

Of course, you might need to change the styles to suite your own needs. This is, obviously, Windows-centric stuff, but if you had a need to do something like this for X11, I'm pretty sure you could use a similar technique.

Later, I make use of this within my other control as such:

```
class TestCtrl : public Upp::Ctrl
{
protected:

 ChildCtrl m_wnd;
public:
 TestCtrl()
 {
 };

 ~TestCtrl()
 {
 };

 void IncomingMessage( const SomeCommand& command )
 {
  if ( m_wnd.GetHWND() == 0 )
  {
   m_wnd.Initialize( ::GetActiveWindow(), *this );
  }
  ::DoSomeCommand( m_wnd.GetHWND(), command );
 };

 void EnableCtrl( bool enable = true )
 {
  if ( enable )
  {
   m_wnd.Show();
  }
  else
  {
   m_wnd.Hide();
  }
 };
};
```

The only thing not quite working the way I want is the size of window... the HWND seems to think it should appear above where you actually place TestCtrl.

I'm currently working around this by moving the control down well below where I really want it... but I want to fix this when I have some time. It's puzzling the SetRect() seems to be a little inaccurate for HWNDs. I'm guessing that U++ is compensating for something (I will have to dig in the code sometime to see).

Of course, it's also possible that ::DoSomeCommand() is actually putting the video off somewhere, but I kind of doubt it; we use ::DoSomeCommand() elsewhere, and it works just fine.

Using this technique, the control should get deallocated on destruction (our favorite U++ and C++ idiom, from what I've seen). The only awkward thing is that EnableCtrl() bit... I'm having to use this to hide and show the window when necessary. I could probably handle this more elegantly by overriding the Hide/Show commands in TestCtrl() to also hide and show the child control.