

Re: hiding an interface (VB6)

Source: <http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2004-05/0068.html>

From: Ralph (nt_consulting32_at_hotmail.com)

Date: 04/30/04

Date: Fri, 30 Apr 2004 10:29:16 -0500

"Mark Meyers" <mmeyers[at]hydromilling.com> wrote in message
news:uGDXoKrLEHA.2456@TK2MSFTNGP12.phx.gbl...

> Using VB 6...

> I have an object (MyObject) that I would like to be able to contain a few
> values, and its reference be passed off to other objects interested in
> receiving it. However; I would like for the recipients to receive only
the

> ability to call the read functions of the object passed, I do not want
them

> to even be able to get to the write members.

>

> And yet, I want on the object owner's side the ability to call the write
> members.

>

> It looks like the only 'friend' I've got in VB is one where the class
become

> public to everything in the project. Is that true? Aside from friend
> access, I don't know how to achieve the desired results.

>

> The closest thing I have come up with is having an Init() function that
> ensures that it is only called once, and where the Init() function is the
> only way to set values in the object. Then, the object recipient could
try

> to call Init(), but that wouldn't work, and so the recipient can only
access

> the read members.

>

> One other idea is to implement two interfaces in MyObject. One with read,
> the other with write, members. The recipients use refs to one interface,
> the owner has refs to the both. The only problem with this, technically,
is

> the recipient still has the ability to knowingly go and establish a
> reference to the other interface, causing it to become publicly accessible
> as well.

>

> - Mark

microsoft.public.vb.general.discussion: Re: hiding an interface (VB6)

One hack is to create an 'owner key'. The owner or creator of your class creates a key when Init() is called. All subsequent calls to the default interface must contain the key and are compared to the original and ignored if the key is not valid. Any object using the 'read' interface doesn't need a key.

-ralph