

Re: MustInherit in Window forms

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2004-07/5797.html>

From: Tom Dacon (*tdacon_at_community.nospam*)

Date: 07/24/04

Date: Fri, 23 Jul 2004 21:36:46 -0700

This is good.

Tom

"Nick Hall" <nickh@aslan.nospam.co.uk> wrote in message
news:eQunzILcEHA.3420@TK2MSFTNGP10.phx.gbl...

> Niklas,

>

> *The way I've done this before is to use good old conditional compilation
> like so*

>

> *#If Debug then*

> *Public Class BaseClass*

> *#Else*

> *Public MustInherit BaseClass*

> *#End If*

>

> *#If Debug Then*

> *Protected Sub OverridableMethod()*

> *Throw New InvalidOperationException("Method not overridden!")*

> *End Sub*

> *#Else*

> *Protected MustOverride Sub OverridableMethod()*

> *#End if*

>

>

> *This means you get a runtime exception in debug builds if you call the
> method BUT you can use the Designer, or a compile time error when you
> compile in Release mode.*

>

> *Hope this helps,*

>

> *Nick Hall*

>

> *"Niklas" <Niklas@discussions.microsoft.com> wrote in message*

> *news:02641378-F4C8-4C9F-9932-CFA3AEA53DBD@microsoft.com...*

> *Hi*

> > *Thank you. I understand the concepts and with the knowledge I have, I
> wanted to create a base form which forced the developers which derives
from
> the Base form to implement DisconnectFromAbrServer(). My way of doing it
> created problems in Visual Studio, see old message below. When I try to
open
> the derived form in VS I get the error message "...The designr need to
> create a instance of type 'WindowApplication1.frmBaseMustInherit' but it
can
> not because the type is declared as abstract". If I run the code the
derived
> form is displayed. It seems to me that Visual Studio 2003 do not allow
forms
> to be MustInherit! Can I force developers to not to forget to implement
the
> method DisconnectFromAbrServer() in another way, now when MustInherit and
> MustOverride do not work in VS?
> > *Regards*
> > */Niklas*
> >
> > *"Tom Dacon" wrote:*
> >
> > > *Oops...*
> > >
> > > so you can choose to code a generic DisconnectFromAbrServer()
> method
> > > there, and it'll be run if the SUBCLASS implementor chooses not to
> implement
> > > a custom version of the method. ...
> > >
> > > *Sorry 'bout that.*
> > >
> > > *Tom Dacon*
> > > *Dacon Software Consulting*
> > >
> > > *"Tom Dacon" <tdacon@community.nospam> wrote in message*
> > > *news:OICdImAcEHA.2388@TK2MSFTNGP11.phx.gbl...*
> > > > *The essence of a MustInherit form is that it's an abstract form: it
> can't
> > > > itself be instantiated. Think of an abstract Automobile class,
marked
> > > > MustInherit, with subclasses such as Ford Explorer, Mazda Miata,
etc.
> > > > You
> > > > can't just walk into an automobile dealership and buy an Automobile.
> > > > You
> > > > have to buy an instance of an instantiable subclass such as a Ford
> > > > Explorer.
> > > > An abstract class can be thought of as a template or a design
pattern,
> > > > sort**

> > > > of, and is most useful as a mechanism for polymorphism: that is, you
> can
> > > > downcast any instance of a subclass to the base class and invoke one
> of a
> > > > known list of members that are available for all subclasses of the
> base
> > > > class, while actually running the code that was written for the
> subclass
> > > > (although for completeness' sake I must note that you can force the
> base
> > > > class's method be invoked, if you choose).
> > > >
> > > > Consequently, much depends on whether your base class is usable
> itself:
> > > that
> > > > is, do you want applications to be able to just create and use an
> instance
> > > > of the base class as a 'concrete class', or must they subclass it
with
> > > > custom logic in all cases?
> > > >
> > > > If you don't want your base class to be instantiated, mark it
> MustInherit.
> > > > Even in a MustInherit class, you can supply default implementations
of
> > > > properties and methods, so you can choose to code a generic
> > > > DisconnectFromAbrServer() method there, and it'll be run if the base
> class
> > > > implementor chooses not to implement a custom version of the method.
> You
> > > can
> > > > force it to be run in all cases, or you can allow it to be
overridden
> in
> > > the
> > > > subclass through the overridable keyword.
> > > >
> > > > If you are OK with having your base class instantiated, you don't
mark
> it
> > > > MustInherit. If a generic implementation of
DisconnectFromAbrServer()
> > > method
> > > > will suffice for all subclasses if no other is available, you can
> > > implement
> > > > it in the base class. If you are OK with subclasses supplying their
> own
> > > > implementation, mark the method Overridable. If you REQUIRE that
they
> > > > implement their own (i.e., a generic implementation cannot be used),
> mark

> > > *the*
> > > *method MustOverride and don't supply any code in the base class.*
> > >
> > > *And document, document, document...*
> > >
> > > *This whole inheritance business, what with new, shadows, overrides,*
> > > *overridable, mustoverride, mustinherit, etc., is pretty complex. I'd*
> > > *suggest*
> > > *taking the time to write a testcase that exercises the various*
> > > *combinations*
> > > *so that you can study the nuances. I've done this for myself in C#,*
> *but*
> > > *haven't yet taken the time to do the same for VB, otherwise I'd post*
> *the*
> > > *testcase.*
> > >
> > > *HTH,*
> > > *Tom Dacon*
> > > *Dacon Software Consulting*
> > >
> > >
> > > *"Niklas" <Niklas@discussions.microsoft.com> wrote in message*
> > > *news:7DF800BC-EDD9-470E-BA70-643888034AE7@microsoft.com...*
> > > > *Hi*
> > > > *I have made a base class and to be sure that developers do not*
> *forget to*
> > > > *implement DisconnectFromAbrServer I have done the following:*
> > > >
> > > > > *Friend MustInherit Class frmBaseAbrForm*
> > > > > *Inherits System.Windows.Forms.Form*
> > > > > *Implements IAbrServerEnabled*
> > > > > *...*
> > > > > *Friend MustOverride Sub DisconnectFromAbrServer() Implements*
> > > > > *IAbrServerEnabled.DisconnectFromAbrServer*
> > > > >
> > > > > *In the derived class:*
> > > > >
> > > > > > *Friend Class frmNewTocc*
> > > > > > *Inherits frmBaseAbrForm*
> > > > > > *...*
> > > > > > *Friend Overrides Sub DisconnectFromAbrServer()*
> > > > > > *'TODO: Add code.*
> > > > > > *End Sub*
> > > > > >
> > > > > > *The problem is that the VS designer can not create an instance of*
> > > > > > *frmBaseAbrForm because it is declared as abstract and an exception*
is
> > > > *thrown*
> > > > *in design time. What can I do? Is it not possible to create Forms as*
> > > > *MustInherit?*
> > > > >

> > > > *Regards*

> > > > */Niklas*

> > > >

> > > >

> > > >

> > >

> > >

> > >

>

>