
Subject: Re: singleton usage

Posted by [Antonio Santiago](#) on Tue, 21 Feb 2006 13:58:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

vcarlos wrote:

> Hi all,

>

> maybe it is the dumbiest question of all time about singleton, but

> since I am in doubt, is better ask than keep it :)

>

> When you define a singleton object, everywhere in the program you use

> it you must call something like this: myObject = Obj_New('Singleton'),

> and then myObject will point to the same instance created before. That

> is fine to me, but what I have doubt is about the usage of this feature

> in my program, since everytime I want access that object I will need to

> write the same statement above, or set an attribute in a class to point

> to the singleton instance, right?

>

> What if, instead of doing that, I create an read only system variable

> holding the object that I want use over the program?

>

One of posible implementations of a Singleton in IDL uses a system

variable to hold the object reference. Other solution uses a common block.

> I know I lost some features, but is not it more useful?

>

> Well, that is it. Thanks

>

> (sorry about my english)

>

> Vinicius

>

--

Antonio Santiago Piñerez

(email: santiago@grahi.upc.edu)

(www: <http://www.grahi.upc.edu/santiago>)

(www: <http://asantiago.blogspot.org>)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia

Universitat Politècnica de Catalunya

Subject: Re: singleton usage

Posted by [vcarlos](#) on Tue, 21 Feb 2006 14:06:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Antonio,

I saw the system variable singleton implementation, but what I am in doubt is about the usage of the pattern itself. Since everytime I need a reference to the singleton object in my objects I need to call a `myObject = Singleton()` statement, or initialize an attribute in every object needing a reference to the singleton. But what if I just create a read only system variable and use it over my whole program? (this variable would be an instance of my "singleton" class)

Thanks

Vinicius

Subject: Re: singleton usage

Posted by [Antonio Santiago](#) on Tue, 21 Feb 2006 14:47:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

vcarlos wrote:

> Hi Antonio,
>
> I saw the system variable singleton implementation, but what I am in
> doubt is about the usage of the pattern itself. Since everytime I need
> a reference to the singleton object in my objects I need to call a
> `myObject = Singleton()` statement, or initialize an attribute in every
> object needing a reference to the singleton. But what if I just create
> a read only system variable and use it over my whole program? (this
> variable would be an instance of my "singleton" class)
>
> Thanks
>
> Vinicius
>

Ok sorry, I don't understand you in the first email.

Yes, I think you can do it, but this is a convention that you use in your program. Then you need to document very well that variable XXX is a "reference" to a singleton object in your whole application.

--

Antonio Santiago Píez
(email: santiago@grahi.upc.edu)

(www: <http://www.grahi.upc.edu/santiago>)
(www: <http://asantiago.blogspot.org>)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia
Universitat Politècnica de Catalunya

Subject: Re: singleton usage
Posted by [JD Smith](#) on Tue, 21 Feb 2006 18:28:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 21 Feb 2006 15:47:26 +0100, Antonio Santiago wrote:

> vcarlos wrote:
>> Hi Antonio,
>>
>> I saw the system variable singleton implementation, but what I am in
>> doubt is about the usage of the pattern itself. Since everytime I need a
>> reference to the singleton object in my objects I need to call a
>> myObject = Singleton() statement, or initialize an attribute in every
>> object needing a reference to the singleton. But what if I just create a
>> read only system variable and use it over my whole program? (this
>> variable would be an instance of my "singleton" class)
>>
> Ok sorry, I don't understand you in the first email. Yes, I think you can
> do it, but this is a convention that you use in your program. Then you
> need to document very well that variable XXX is a "reference" to a
> singleton object in your whole application.

Using a singleton generator is easy (see various threads dating back several years). As far as preventing rogue "obj_new()" calls, you can also check the common block or system variable explicitly in the Init function and generate an error if anyone uses it directly (except for the first call).

JD
