

---

Subject: Re: Catalyst CatDestroyDefaults

Posted by [David Fanning](#) on Tue, 23 Jun 2009 13:26:05 GMT

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

---

Crane writes:

> I am trying to use CatSetDefault to set some of my system settings,  
> for instance:  
>  
> oLang = Obj\_New('languageclass')  
> CatSetDefault, 'language', oLang  
>  
> When my application is destroyed, the CatDestroyDefaults is called  
> automatically, I expected this procedure will clean up all system  
> settings, so I did not destroy the oLang object by myself.  
> Unfortunately, oLang is not cleaned up, it is still in memory. Should  
> I have to explicitly destroy the system settings in cleanup method ,  
> or where am I wrong?  
>  
> I try to find what happens using debug mode, and know the  
> CatDestroyDefaults procedure finally go to the CatContainer::Cleanup  
> method, this method remove all child objects so they don't get  
> destroyed automatically, I think this is why oLang is not destroy.  
>  
> I do not understand this, because we need CatDestroyDefaults to  
> automatically destroy all the system settings when we destroy the  
> application.

It is probably appropriate that this comes on the heels of another discussion a week or so ago about the inability of UNDEFINE to undefine everything. And the reason for the problem is similar. It never occurred to me to store a heap variable in a Catalyst default value. ;-)

But, now I know someone wants to do that, the solution is simple. If the internal pointer points to a heap variable, destroy that, too. And since I already spent an hour getting UNDEFINE to do just that a week ago, I decided to use that to get the job done.

So, be sure you have the latest UNDEFINE, and you can download a new CatListValue object here:

<http://www.dfanning.com/programs/undefine.pro>  
.../programs/catalyst/source/core/catlistvalue\_\_define.pro

Tested only lightly, as I am late for work, as usual. :-)

Cheers,

David

--

David Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: Catalyst CatDestroyDefaults  
Posted by [David Fanning](#) on Sun, 28 Jun 2009 14:09:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning writes:

- > It is probably appropriate that this comes on the heels of
- > another discussion a week or so ago about the inability of
- > UNDEFINE to undefine everything. And the reason for the problem
- > is similar. It never occurred to me to store a heap variable
- > in a Catalyst default value. ;-)
- >
- > But, now I know someone wants to do that, the solution is simple.
- > If the internal pointer points to a heap variable, destroy that, too.
- > And since I already spent an hour getting UNDEFINE to do just that
- > a week ago, I decided to use that to get the job done.
- >
- > So, be sure you have the latest UNDEFINE, and you can download
- > a new CatListValue object here:
- >
- > <http://www.dfanning.com/programs/undefine.pro>
- > [.../programs/catalyst/source/core/catlistvalue\\_\\_define.pro](http://www.dfanning.com/programs/catalyst/source/core/catlistvalue__define.pro)
- >
- > Tested only lightly, as I am late for work, as usual. :-)

Whoops! Should have tested this more extensively for the Law of Unintended Consequences. :-(

It was a good idea, in theory, but it had trickle down effects in other parts of Catalyst. The most sensible course seems to me to keep the underlying behavior the same, but to allow different behavior by means of a keyword. (Almost \*always\* the most sensible course, but I do not always think clearly before the coffee kicks in.)

Thus, three Catalyst routines have changed to support a DESTROY\_ON\_CLEANUP keyword to the CatListValue object. If this keyword is set, the IDL variable stored in the

CatListValue storage location is destroyed with UNDEFINE when the CatListValue object is destroyed.

The same keyword is available now for CatSetDefault, so that you can destroy variables when default variables are cleaned up. The following code will prevent memory leakage when storing an object as a Catalyst default value.

```
pb = Obj_New('ProgressBar')
CatSetDefault, 'ProgressBar', pb, /DESTROY_ON_CLEANUP
CatDestroyDefaults
```

The following three routines have changed:

```
.../catalyst/source/utilities/catsetdefault.pro
.../catalyst/source/core/catlist__define.pro
.../catalyst/source/core/catlistvalue__define.pro
```

Documentation of this keyword is pretty extensive, but please ask questions if you are not clear about it.

Cheers,

David

--

David Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---