
Subject: Renegade C++ Cleanup
Posted by [halo2pac](#) on Mon, 27 Sep 2010 23:31:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hey guys.

I'm working on a lot of C++ that has to do with renegade in general (servers, scripts, ect).

When you write code in let say a script what do you have to delete to make sure you don't have any memory leaks?

So far I know of when you create a string like:
const char *Nick

you have to delete it.

what other things are there to delete, and how do I delete them?

*off the top of my head Vector3 is one of the ones where I am wondering if I have to delete it after creating one.

Thanks!

Subject: Re: Renegade C++ Cleanup
Posted by [jonwil](#) on Tue, 28 Sep 2010 02:19:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Basically anywhere you see something like new char[100], thats allocating memory that you need to clean up.

If you do something like Vector3 v;

v.X = 1;

v.Y = 1;

v.Z = 1;

then, no, you dont have to clean up that Vector3.

Subject: Re: Renegade C++ Cleanup
Posted by [halo2pac](#) on Tue, 28 Sep 2010 13:39:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Then I have a 1000 memory leaked lol. How do u clean up a char test[[]]?

Subject: Re: Renegade C++ Cleanup

Posted by [Omar007](#) on Tue, 28 Sep 2010 14:18:21 GMT

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

afaik just do
delete test;

But to be honest I never deleted that stuff in my scripts.. (yet)

Subject: Re: Renegade C++ Cleanup

Posted by [jonwil](#) on Tue, 28 Sep 2010 16:19:39 GMT

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

If you have something like this

```
char test[100];
```

then you dont need to do anything to delete it, it gets deleted automatically.

Subject: Re: Renegade C++ Cleanup

Posted by [saberhawk](#) on Tue, 28 Sep 2010 17:29:29 GMT

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

The rules basically are:

If you called new on it, use delete on it when you no longer need it.

If you called new[] on it (ie char* test = new char[128], use delete[] on it.

Debug mode builds enable the memory manager code in 3.4.4. iirc not all of the "leaks" listed can be fixed, but they do show where allocations occurred. Certain leaks were fixed in code released by me not included with 3.4.4 (notably a SurfaceClass leak that sometimes causes lag on exit) and the changes themselves were likely also documented in a post of mine.

Subject: Re: Renegade C++ Cleanup

Posted by [cAmpa](#) on Tue, 28 Sep 2010 17:49:19 GMT

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

I think i don't know all your fixes, could you show me all?

Subject: Re: Renegade C++ Cleanup

Posted by [saberhawk](#) on Tue, 28 Sep 2010 17:59:41 GMT

cAmpa wrote on Tue, 28 September 2010 13:49 I think i don't know all your fixes, could you show me all?

Gladly; I just need you to show me all the fixes I've made. I don't actually remember all the changes I've made, searching for posts made by me is going to give you better results. One thing I'm pretty sure of is that newer posts I've made including code still contain the fixes of earlier posts.
