

Re: [RFC] kobject: add kobject_init_ng and kobject_init_and_add functions

Re: [RFC] kobject: add kobject_init_ng and kobject_init_and_add functions

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-11/msg10323.html>

- *From:* Greg KH <greg@xxxxxxxxx>
 - *Date:* Fri, 30 Nov 2007 13:48:19 -0800
-

On Fri, Nov 30, 2007 at 04:19:53PM -0500, Alan Stern wrote:

On Fri, 30 Nov 2007, Greg KH wrote:

My suggestion: Have kobject_init_ng() accept a ktype pointer but not a parent or name. Instead, make kobject_add_ng() take the parent and name (possibly a kset also). Then when kobject_init_and_add() encounters an error, it shouldn't do a _put() -- the caller can either do the _put() or just do a kfree().

Why not the parent for init()? Isn't it always known at that time? I'll dig to be sure.

Specifying the parent during _add() is more logical, because a kobject doesn't actually _do_ anything to the parent until it is registered in the parent's directory. Or to put it another way, an unregistered kobject can't have a parent in any meaningful sense so there's no point specifying the parent in the _init() call.

Ok, how about this:

```
void kobject_init(struct kobject *kobj, struct ktype *ktype);
```

and then:

```
int kobject_add(struct kobject *kobj, struct kobject *parent, const char *fmt, ...);
```

After we call kobject_init() we HAVE to call kobject_put() to clean up properly. So, if kobject_add() fails, we still need to clean up with kobject_put();

That means we _can_ create a:

Re: [RFC] kobject: add kobject_init_ng and kobject_init_and_add functions

Re: [RFC] kobject: add kobject_init_ng and kobject_init_and_add functions

```
int kobject_init_and_add(struct kobject *kobj, struct ktype *ktype, struct kobject *parent, const char *fmt, ...);
```

and if that fails, then again, you have to call kobject_put() to clean things up, right?

Does this look sane?

thanks,

greg k-h

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>