
Subject: Re: sysfs : fix kobject rename with multiple namespaces

Posted by [serue](#) on Mon, 05 May 2008 16:56:15 GMT

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

Quoting Daniel Lezcano (dlezcano@fr.ibm.com):

> To be applied on top of the sysfs per namespace patchset.

> Subject: avoid kobject name conflict with different namespaces

> From: Daniel Lezcano <dlezcano@fr.ibm.com>

>

> The renaming of a kobject will fail if there is another kobject

> with the same name belonging to another namespace.

>

> This patch makes the kobject lookup in kobject_rename to check if

> the object exists _and_ belongs to the same namespace.

>

> Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

> ---

> fs/sysfs/dir.c | 10 ++++++++

> include/linux/sysfs.h | 7 ++++++

> lib/kobject.c | 2 +-
> 3 files changed, 18 insertions(+), 1 deletion(-)

>

> Index: net-next-2.6/fs/sysfs/dir.c

> =====

> --- net-next-2.6.orig/fs/sysfs/dir.c

> +++ net-next-2.6/fs/sysfs/dir.c

> @@ -899,6 +899,16 @@ err_out:

> return error;

> }

>

> +int sysfs_tag_cmp(struct kobject * kobj1, struct kobject *kobj2)

The patch looks good to me, except I'd really rename this to 'sysfs_tag_eq'. Calling it 'cmp', returning 't1!=t2', then checking for !sysfs_tag_cmp really increases the amount of thinking i have to do to make sure everything is kosher :)

Maybe it's just me.

thanks,

-serge

> +{

> + struct sysfs_dirent *sd1 = kobj1->sd;

> + struct sysfs_dirent *sd2 = kobj2->sd;

> + const void *tag1 = sysfs_dirent_tag(sd1);

> + const void *tag2 = sysfs_dirent_tag(sd2);

```

> +
> + return tag1 != tag2;
> +}
> +
> int sysfs_rename_dir(struct kobject * kobj, const char *new_name)
> {
>     struct sysfs_dirent *sd = kobj->sd;
> Index: net-next-2.6/include/linux/sysfs.h
> =====
> --- net-next-2.6.orig/include/linux/sysfs.h
> +++ net-next-2.6/include/linux/sysfs.h
> @@ -95,6 +95,8 @@ int sysfs_schedule_callback(struct kobje
>
> int __must_check sysfs_create_dir(struct kobject *kobj);
> void sysfs_remove_dir(struct kobject *kobj);
> +int sysfs_tag_cmp(struct kobject * kobj1, struct kobject *kobj2);
> +
> int __must_check sysfs_rename_dir(struct kobject *kobj, const char *new_name);
> int __must_check sysfs_move_dir(struct kobject *kobj,
>     struct kobject *new_parent_kobj);
> @@ -152,6 +154,11 @@ static inline void sysfs_remove_dir(stru
> ;
> }
>
> +static inline int sysfs_tag_cmp(struct kobject * kobj1, struct kobject *kobj2)
> +{
> + return 0;
> +}
> +
> static inline int sysfs_rename_dir(struct kobject *kobj, const char *new_name)
> {
>     return 0;
> Index: net-next-2.6/lib/kobject.c
> =====
> --- net-next-2.6.orig/lib/kobject.c
> +++ net-next-2.6/lib/kobject.c
> @@ -418,7 +418,7 @@ int kobject_rename(struct kobject *kobj,
>     if (kobj->kset) {
>         struct kobject *temp_kobj;
>         temp_kobj = kset_find_obj(kobj->kset, new_name);
> - if (temp_kobj) {
> + if (temp_kobj && !sysfs_tag_cmp(temp_kobj, kobj)) {
>     printk(KERN_WARNING "kobject '%s' cannot be renamed "
>         "to '%s' as '%s' is already in existence.\n",
>         kobject_name(kobj), new_name, new_name);

```

Containers mailing list

