

---

Subject: [PATCH 01/15] kobject: Cleanup kobject\_rename and !CONFIG\_SYSFS  
Posted by [ebiederm](#) on Fri, 04 Jul 2008 01:05:28 GMT

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

---

It finally dawned on me what the clean fix to sysfs\_rename\_dir calling kobject\_set\_name is. Move the work into kobject\_rename where it belongs. The callers serialize us anyway so this is safe.

Signed-off-by: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)>

---

```
fs/sysfs/dir.c      | 6 +-----
include/linux/sysfs.h | 4 +---
lib/kobject.c       | 17 ++++++++-----
3 files changed, 17 insertions(+), 10 deletions(-)
```

```
diff --git a/fs/sysfs/dir.c b/fs/sysfs/dir.c
index 8c0e4b9..146b86a 100644
```

```
--- a/fs/sysfs/dir.c
```

```
+++ b/fs/sysfs/dir.c
```

```
@@ -799,16 +799,12 @@ int sysfs_rename_dir(struct kobject * kobj, const char *new_name)
    if (!new_dentry)
        goto out_unlock;
```

```
- /* rename kobject and sysfs_dirent */
```

```
+ /* rename sysfs_dirent */
```

```
    error = -ENOMEM;
```

```
    new_name = dup_name = kstrdup(new_name, GFP_KERNEL);
```

```
    if (!new_name)
```

```
        goto out_unlock;
```

```
- error = kobject_set_name(kobj, "%s", new_name);
```

```
- if (error)
```

```
- goto out_unlock;
```

```
-
```

```
    dup_name = sd->s_name;
```

```
    sd->s_name = new_name;
```

```
diff --git a/include/linux/sysfs.h b/include/linux/sysfs.h
```

```
index 84d92bb..f7e43ed 100644
```

```
--- a/include/linux/sysfs.h
```

```
+++ b/include/linux/sysfs.h
```

```
@@ -20,8 +20,6 @@
```

```
struct kobject;
```

```
struct module;
```

```
-extern int kobject_set_name(struct kobject *kobj, const char *name, ...)
```

```
-    __attribute__((format(printf, 2, 3)));
```

```

/* FIXME
 * The *owner field is no longer used, but leave around
 * until the tree gets cleaned up fully.
@@ -140,7 +138,7 @@ static inline void sysfs_remove_dir(struct kobject *kobj)

static inline int sysfs_rename_dir(struct kobject *kobj, const char *new_name)
{
- return kobject_set_name(kobj, "%s", new_name);
+ return 0;
}

static inline int sysfs_move_dir(struct kobject *kobj,
diff --git a/lib/kobject.c b/lib/kobject.c
index 829b839..49b3bc4 100644
--- a/lib/kobject.c
+++ b/lib/kobject.c
@@ -451,6 +451,7 @@ int kobject_rename(struct kobject *kobj, const char *new_name)
{
    int error = 0;
    const char *devpath = NULL;
+ const char *dup_name = NULL, *name;
    char *devpath_string = NULL;
    char *envp[2];

@@ -474,15 +475,27 @@ int kobject_rename(struct kobject *kobj, const char *new_name)
    envp[0] = devpath_string;
    envp[1] = NULL;

+ name = dup_name = kstrdup(new_name, GFP_KERNEL);
+ if (!name) {
+     error = -ENOMEM;
+     goto out;
+ }
+
    error = sysfs_rename_dir(kobj, new_name);
+ if (error)
+     goto out;
+
+ /* Install the new kobject name */
+ dup_name = kobj->name;
+ kobj->name = name;

/* This function is mostly/only used for network interface.
 * Some hotplug package track interfaces by their name and
 * therefore want to know when the name is changed by the user. */
- if (!error)
-     kobject_uevent_env(kobj, KOBJ_MOVE, envp);
+ kobject_uevent_env(kobj, KOBJ_MOVE, envp);

```

```
out:
+ kfree(dup_name);
  kfree(devpath_string);
  kfree(devpath);
  kobject_put(kobj);
--
1.5.3.rc6.17.g1911
```

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---