
Subject: [PATCH 08/11] driver core: Implement tagged directory support for device classes.

Posted by [Benjamin Thery](#) on Wed, 18 Jun 2008 17:08:49 GMT

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

driver core: Implement tagged directory support for device classes.

This patch enables tagging on every class directory if struct class has tag_ops.

In addition device_del and device_rename were modified to use sysfs_delete_link and sysfs_rename_link respectively to ensure when these operations happen on devices whose classes have tag_ops that they work properly.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>

```
drivers/base/class.c | 30 ++++++-----
drivers/base/core.c | 41 ++++++-----
include/linux/device.h | 2 ++
3 files changed, 51 insertions(+), 22 deletions(-)
```

Index: linux-mm/drivers/base/class.c

=====

--- linux-mm.orig/drivers/base/class.c

+++ linux-mm/drivers/base/class.c

```
@@ -135,6 +135,17 @@ static void remove_class_attrs(struct cl
 }
 }
```

```
+static int class_setup_tagging(struct class *cls)
+{
+ const struct sysfs_tagged_dir_operations *tag_ops;
+
+ tag_ops = cls->tag_ops;
+ if (!tag_ops)
+ return 0;
+
+ return sysfs_enable_tagging(&cls->p->class_subsys.kobj, tag_ops);
+}
+
+int __class_register(struct class *cls, struct lock_class_key *key)
+{
+ struct class_private *cp;
@@ -171,13 +182,24 @@ int __class_register(struct class *cls,
 cls->p = cp;
```

```

    error = kset_register(&cp->class_subsys);
- if (error) {
-     kfree(cp);
-     return error;
- }
+ if (error)
+     goto out_free_cp;
+
+ error = class_setup_tagging(cls);
+ if (error)
+     goto out_unregister;
+
+ error = add_class_attrs(class_get(cls));
+ class_put(cls);
+ if (error)
+     goto out_unregister;
+out:
+     return error;
+out_unregister:
+ kset_unregister(&cp->class_subsys);
+out_free_cp:
+ kfree(cp);
+ goto out;
}
EXPORT_SYMBOL_GPL(__class_register);

```

Index: linux-mm/drivers/base/core.c

```

=====
--- linux-mm.orig/drivers/base/core.c
+++ linux-mm/drivers/base/core.c
@@ -618,6 +618,10 @@ static struct kobject *get_device_parent
     kobject_put(k);
     return NULL;
 }
+ /* If we created a new class-directory setup tagging */
+ if (dev->class->tag_ops)
+     sysfs_enable_tagging(k, dev->class->tag_ops);
+
+ /* do not emit an uevent for this simple "glue" directory */
+ return k;
+ }
@@ -754,13 +758,14 @@ static void device_remove_class_symlinks

if (dev->kobj.parent != &dev->class->p->class_subsys.kobj &&
    device_is_not_partition(dev))
- sysfs_remove_link(&dev->class->p->class_subsys.kobj,
+ sysfs_delete_link(&dev->class->p->class_subsys.kobj, &dev->kobj,
    dev_name(dev));

```

```

#else
if (dev->parent && device_is_not_partition(dev))
    sysfs_remove_link(&dev->kobj, "device");

- sysfs_remove_link(&dev->class->p->class_subsys.kobj, dev_name(dev));
+ sysfs_delete_link(&dev->class->p->class_subsys.kobj, &dev->kobj,
+   dev_name(dev));
#endif

    sysfs_remove_link(&dev->kobj, "subsystem");
@@ -1349,6 +1354,16 @@ int device_rename(struct device *dev, ch
    strcpy(old_device_name, dev_name(dev), BUS_ID_SIZE);
    strcpy(dev->bus_id, new_name, BUS_ID_SIZE);

+ #ifndef CONFIG_SYSFS_DEPRECATED
+ if (dev->class &&
+   (dev->kobj.parent != &dev->class->p->class_subsys.kobj)) {
+   error = sysfs_rename_link(&dev->class->p->class_subsys.kobj,
+   &dev->kobj, old_device_name, new_name);
+   if (error)
+   goto out;
+ }
+ #endif
+
    error = kobject_rename(&dev->kobj, new_name);
    if (error) {
        strcpy(dev->bus_id, old_device_name, BUS_ID_SIZE);
@@ -1357,23 +1372,13 @@ int device_rename(struct device *dev, ch

# ifdef CONFIG_SYSFS_DEPRECATED
    if (old_class_name) {
+ error = -ENOMEM;
        new_class_name = make_class_name(dev->class->name, &dev->kobj);
- if (new_class_name) {
-   error = sysfs_create_link(&dev->parent->kobj,
-   &dev->kobj, new_class_name);
-   if (error)
-   goto out;
-   sysfs_remove_link(&dev->parent->kobj, old_class_name);
- }
- }
- # else
- if (dev->class) {
-   error = sysfs_create_link(&dev->class->p->class_subsys.kobj,
-   &dev->kobj, dev_name(dev));
-   if (error)
-   goto out;
-   sysfs_remove_link(&dev->class->p->class_subsys.kobj,

```

```

-    old_device_name);
+ if (new_class_name)
+ error = sysfs_rename_link(&dev->parent->kobj,
+     &dev->kobj,
+     old_class_name,
+     new_class_name);
+ }
#endif

```

Index: linux-mm/include/linux/device.h

```

=====
--- linux-mm.orig/include/linux/device.h
+++ linux-mm/include/linux/device.h
@@ -195,6 +195,8 @@ struct class {
    int (*suspend)(struct device *dev, pm_message_t state);
    int (*resume)(struct device *dev);

+ const struct sysfs_tagged_dir_operations *tag_ops;
+
    struct pm_ops *pm;
    struct class_private *p;
};

--

```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
