
Subject: [PATCH 5/5] driver core: Implement shadow directory support for device classes.

Posted by [ebiederm](#) on Fri, 06 Apr 2007 17:14:52 GMT

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

This patch enables shadowing on every class directory if struct class has shadow_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 in the presence of shadow directories the work properly.

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

drivers/base/class.c | 30 ++++++++-----

drivers/base/core.c | 20 +++++-----

include/linux/device.h | 2 ++

3 files changed, 39 insertions(+), 13 deletions(-)

diff --git a/drivers/base/class.c b/drivers/base/class.c

index 80bbb20..1ac2b0e 100644

--- a/drivers/base/class.c

+++ b/drivers/base/class.c

```
@@ -136,6 +136,17 @@ static void remove_class_attrs(struct class * cls)
}
}
```

```
+static int class_setup_shadowing(struct class *cls)
```

```
+{
```

```
+ const struct shadow_dir_operations *shadow_ops;
```

```
+
```

```
+ shadow_ops = cls->shadow_ops;
```

```
+ if (!shadow_ops)
```

```
+ return 0;
```

```
+
```

```
+ return sysfs_enable_shadowing(&cls->subsys.kset.kobj, shadow_ops);
```

```
+}
```

```
+
```

```
int class_register(struct class * cls)
```

```
{
```

```
 int error;
```

```
@@ -154,11 +165,22 @@ int class_register(struct class * cls)
```

```
 subsys_set_kset(cls, class_subsys);
```

```
 error = subsystem_register(&cls->subsys);
```

```
- if (!error) {
```

```
- error = add_class_attrs(class_get(cls));
```

```

- class_put(cls);
- }
+ if (error)
+ goto out;
+
+ error = class_setup_shadowing(cls);
+ if (error)
+ goto out_unregister;
+
+ error = add_class_attrs(cls);
+ if (error)
+ goto out_unregister;
+
+out:
    return error;
+out_unregister:
+ subsystem_unregister(&cls->subsys);
+ goto out;
}

```

```

void class_unregister(struct class * cls)

```

```

diff --git a/drivers/base/core.c b/drivers/base/core.c

```

```

index 8650401..3d9c5ff 100644

```

```

--- a/drivers/base/core.c

```

```

+++ b/drivers/base/core.c

```

```

@@ -519,8 +519,14 @@ static struct kobject * get_device_parent(struct device *dev,
    return kobj;

```

```

/* or create a new class-directory at the parent device */

```

```

- return kobject_kset_add_dir(&dev->class->class_dirs,
+ kobj = kobject_kset_add_dir(&dev->class->class_dirs,
    parent_kobj, dev->class->name);
+
+ /* If we created a new class-directory setup shadowing */
+ if (kobj && dev->class->shadow_ops)
+ sysfs_enable_shadowing(kobj, dev->class->shadow_ops);
+
+ return kobj;
}

```

```

if (parent)

```

```

@@ -797,8 +803,8 @@ void device_del(struct device * dev)

```

```

/* If this is not a "fake" compatible device, remove the
 * symlink from the class to the device. */

```

```

if (dev->kobj.parent != &dev->class->subsys.kset.kobj)

```

```

- sysfs_remove_link(&dev->class->subsys.kset.kobj,
- dev->bus_id);
+ sysfs_delete_link(&dev->class->subsys.kset.kobj,

```

```

+    &dev->kobj, dev->bus_id);
    if (parent) {
#ifdef CONFIG_SYSFS_DEPRECATED
        char *class_name = make_class_name(dev->class->name,
@@ -1096,6 +1102,8 @@ int device_rename(struct device *dev, char *new_name)
        goto out_free_old_class;
    }
    strncpy(old_symlink_name, dev->bus_id, BUS_ID_SIZE);
+ sysfs_rename_link(&dev->class->subsys.kset.kobj, &dev->kobj,
+    old_symlink_name, new_name);
    }

    strncpy(dev->bus_id, new_name, BUS_ID_SIZE);
@@ -1113,12 +1121,6 @@ int device_rename(struct device *dev, char *new_name)
    }
#endif

- if (dev->class) {
- sysfs_remove_link(&dev->class->subsys.kset.kobj,
-    old_symlink_name);
- sysfs_create_link(&dev->class->subsys.kset.kobj, &dev->kobj,
-    dev->bus_id);
- }
    put_device(dev);

    kfree(new_class_name);
diff --git a/include/linux/device.h b/include/linux/device.h
index de0e73e..8326067 100644
--- a/include/linux/device.h
+++ b/include/linux/device.h
@@ -199,6 +199,8 @@ struct class {

    int (*suspend)(struct device *, pm_message_t state);
    int (*resume)(struct device *);
+
+ const struct shadow_dir_operations *shadow_ops;
};

extern int __must_check class_register(struct class *);
--
1.5.0.g53756

```

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