
Subject: [PATCH 2/2] Implement shadow directory support for device classes.

Posted by [ebiederm](#) on Wed, 24 Jan 2007 19:37:19 GMT

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

Modify the device class code so that normal manipulations work in the presence of shadow directories. Some of the shadow directory support still needs to be implemented in the implementation of the class but these modifications are sufficient to make that simple.

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

drivers/base/class.c | 34 ++++++

include/linux/device.h | 4 +

2 files changed, 36 insertions(+), 2 deletions(-)

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

index 8bf2ca2..f72ddc0 100644

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

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

@@ -136,6 +136,18 @@ static void remove_class_attrs(struct class * cls)
}
}

+static int class_setup_shadow(struct class *cls)

+{

+ if (!cls->class_device_dparent && !cls->class_follow_link)

+ return 0;

+

+ if (!cls->class_device_dparent || !cls->class_device_dparent)

+ return -EINVAL;

+

+ return sysfs_make_shadowed_dir(&cls->subsys.kset.kobj,

+ cls->class_follow_link);

+}

+

int class_register(struct class * cls)

{

int error;

@@ -154,6 +166,11 @@ int class_register(struct class * cls)

error = subsystem_register(&cls->subsys);

if (!error) {

+ error = class_setup_shadow(cls);

+ if (error)

+ subsystem_unregister(&cls->subsys);

+ }

+ if (!error) {

error = add_class_attrs(class_get(cls));

```

    class_put(cls);
}
@@ -582,6 +599,7 @@ int class_device_add(struct class_device *class_dev)
    struct class *parent_class = NULL;
    struct class_device *parent_class_dev = NULL;
    struct class_interface *class_intf;
+ struct dentry *dparent;
    int error = -EINVAL;

    class_dev = class_device_get(class_dev);
@@ -610,7 +628,11 @@ int class_device_add(struct class_device *class_dev)
    else
        class_dev->kobj.parent = &parent_class->subsys.kset.kobj;

- error = kobject_add(&class_dev->kobj);
+ if (parent_class->class_device_dparent)
+     dparent = parent_class->class_device_dparent(class_dev);
+ else
+     dparent = parent_class->subsys.kset.kobj.dentry;
+ error = kobject_shadow_add(&class_dev->kobj, dparent);
    if (error)
        goto out2;

@@ -841,11 +863,15 @@ int class_device_rename(struct class_device *class_dev, char
*new_name)
{
    int error = 0;
    char *old_class_name = NULL, *new_class_name = NULL;
+ struct class *class;
+ struct dentry *new_parent;

    class_dev = class_device_get(class_dev);
    if (!class_dev)
        return -EINVAL;

+ class = class_dev->class;
+
    pr_debug("CLASS: renaming '%s' to '%s'\n", class_dev->class_id,
        new_name);

@@ -857,7 +883,11 @@ int class_device_rename(struct class_device *class_dev, char
*new_name)

    strcpy(class_dev->class_id, new_name, KOBJ_NAME_LEN);

- error = kobject_rename(&class_dev->kobj, new_name);
+ if (class->class_device_dparent)
+     new_parent = class->class_device_dparent(class_dev);

```

```

+ else
+ new_parent = class->subsys.kset.kobj.dentry;
+ error = kobject_shadow_rename(&class_dev->kobj, new_parent, new_name);

#ifdef CONFIG_SYSFS_DEPRECATED
    if (class_dev->dev) {
diff --git a/include/linux/device.h b/include/linux/device.h
index f44247f..162e840 100644
--- a/include/linux/device.h
+++ b/include/linux/device.h
@@ -33,6 +33,7 @@ struct device;
struct device_driver;
struct class;
struct class_device;
+struct nameidata;

struct bus_type {
    const char * name;
@@ -197,6 +198,9 @@ struct class {

    int (*suspend)(struct device *, pm_message_t state);
    int (*resume)(struct device *);
+
+ struct dentry *(*class_device_dparent)(struct class_device *);
+ void *(*class_follow_link)(struct dentry *dentry, struct nameidata *nd);
};

extern int __must_check class_register(struct class *);
--
1.4.4.1.g278f

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
