Subject: [PATCH 09/10] driver core: Implement tagged directory support for device classes.
Posted by ebiederm on Sat, 01 Dec 2007 09:33:11 GMT
View Forum Message <> Reply to Message

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>
---
 drivers/base/class.c | 30 ++++++++++++++++++++---
 drivers/base/core.c | 51 ++++++++++++++++++++++++++++++++---------
 include/linux/device.h | 2 +
 3 files changed, 55 insertions(+), 28 deletions(-)

diff --git a/drivers/base/class.c b/drivers/base/class.c
index c4f8843..ed9393d 100644
--- a/drivers/base/class.c
+++ b/drivers/base/class.c
@@ -135,6 +135,17 @@ static void remove_class_attrs(struct class * cls)
  }
 }

+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->subsys.kobj, tag_ops);
+}
+
 int class_register(struct class * cls)
 {
  int error;
@@ -160,11 +171,22 @@ int class_register(struct class * cls)
  cls->subsys.kobj.ktype = &class_ktype;

  error = kset_register(&cls->subsys);
- if (!error) {
+
+

```
-  error = add_class_attrs(class_get(cls));
-  class_put(cls);
- }
+ if (error)
+  goto out;
+
+ error = class_setup_tagging(cls);
+ if (error)
+  goto out_unregister;
+
+ error = add_class_attrs(cls);
+ if (error)
+  goto out_unregister;
+
+out:
  return error;
+out_unregister:
+ kset_unregister(&cls->subsys);
+ goto out;
 }

 void class_unregister(struct class * cls)
diff --git a/drivers/base/core.c b/drivers/base/core.c
index a2c3d4e..f9d3fcf 100644
--- a/drivers/base/core.c
+++ b/drivers/base/core.c
@@ -600,16 +600,20 @@ static struct kobject *get_device_parent(struct device *dev,
    return kobj;

   /* or create a new class-directory at the parent device */
-  k = kobject_create(dev->class->name, parent_kobj);
-  if (!k)
+bser  kobj = kobject_create(dev->class->name, parent_kobj);
+  if (!kobj)
    return NULL;
-  k->kset = &dev->class->class_dirs;
-  retval = kobject_register(k);
+  kobj->kset = &dev->class->class_dirs;
+  retval = kobject_register(kobj);
   if (retval < 0) {
-   kfree(k);
+   kfree(kobj);
    return NULL;
   }
-  return k;
+  /* If we created a new class-directory setup tagging */
+  if (kobj && dev->class->tag_ops)
+   sysfs_enable_tagging(k, dev->class->tag_ops);
```

```
+
+  return kobj;
  }

  if (parent)
@@ -758,7 +762,8 @@ static void device_remove_class_symlinks(struct device *dev)

  if (dev->kobj.parent != &dev->class->subsys.kobj &&
      dev->type != &part_type)
-  sysfs_remove_link(&dev->class->subsys.kobj, dev->bus_id);
+  sysfs_delete_link(&dev->class->subsys.kobj,
+       &dev->kobj, dev->bus_id);
 #else
  if (dev->parent && dev->type != &part_type)
   sysfs_remove_link(&dev->kobj, "device");
@@ -1223,6 +1228,15 @@ int device_rename(struct device *dev, char *new_name)
  strlcpy(old_device_name, dev->bus_id, BUS_ID_SIZE);
  strlcpy(dev->bus_id, new_name, BUS_ID_SIZE);

+#ifndef CONFIG_SYSFS_DEPRECATED
+ if (dev->class && (dev->kobj.parent != &dev->class->subsys.kobj)) {
+  error = sysfs_rename_link(&dev->class->subsys.kobj,
+    &dev->kobj, old_device_name, new_name);
+  if (error)
+   goto out;
+ }
+#endif
+
  error = kobject_rename(&dev->kobj, new_name);
  if (error) {
   strlcpy(dev->bus_id, old_device_name, BUS_ID_SIZE);
@@ -1231,24 +1245,13 @@ int device_rename(struct device *dev, char *new_name)

 #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) {
-  sysfs_remove_link(&dev->class->subsys.kobj, old_device_name);
```

```
-  error = sysfs_create_link(&dev->class->subsys.kobj, &dev->kobj,
-       dev->bus_id);
-  if (error) {
-   dev_err(dev, "%s: sysfs_create_symlink failed (%d)\n",
-    __FUNCTION__, error);
-  }
+  if (new_class_name)
+   error = sysfs_rename_link(&dev->parent->kobj,
+        &dev->kobj,
+        old_class_name,
+        new_class_name);
  }
 #endif

diff --git a/include/linux/device.h b/include/linux/device.h
index ed38712..80ba08f 100644
--- a/include/linux/device.h
+++ b/include/linux/device.h
@@ -187,6 +187,8 @@ struct class {

  int (*suspend)(struct device *, pm_message_t state);
  int (*resume)(struct device *);
+
+ const struct sysfs_tagged_dir_operations *tag_ops;
 };

 extern int __must_check class_register(struct class *);
--
1.5.3.rc6.17.g1911


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