
Subject: [PATCH] net: Fix running without sysfs
Posted by [ebiederm](#) on Wed, 26 Sep 2007 23:48:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

When sysfs support is compiled out the kernel still keeps and maintains the kobject tree. So it is not safe to skip our kobject reference counting or to avoid becoming members of the kobject tree. It is safe to not add the networking specific sysfs attributes.

This patch removes the sysfs special cases from net/core/dev.c renames functions from netdev_sysfs_xxxx to netdev_kobject_xxxx and always compiles in net-sysfs.c

net-sysfs.c is modified with a CONFIG_SYSFS guard around the parts that are actually sysfs specific.

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

```
net/core/Makefile | 2 +-
net/core/dev.c    | 28 ++++++++-----
net/core/net-sysfs.c | 13 ++++++++---
3 files changed, 20 insertions(+), 23 deletions(-)
```

```
diff --git a/net/core/Makefile b/net/core/Makefile
```

```
index ea9b3f3..b1332f6 100644
```

```
--- a/net/core/Makefile
```

```
+++ b/net/core/Makefile
```

```
@@ -11,7 +11,7 @@ obj-y += dev.o ethtool.o dev_mcast.o dst.o netevent.o \
    neighbour.o rtnetlink.o utils.o link_watch.o filter.o
```

```
obj-$(CONFIG_XFRM) += flow.o
```

```
-obj-$(CONFIG_SYSFS) += net-sysfs.o
```

```
+obj-y += net-sysfs.o
```

```
obj-$(CONFIG_NET_PKTGEN) += pktgen.o
```

```
obj-$(CONFIG_NETPOLL) += netpoll.o
```

```
obj-$(CONFIG_NET_DMA) += user_dma.o
```

```
diff --git a/net/core/dev.c b/net/core/dev.c
```

```
index 91c31e6..a99225e 100644
```

```
--- a/net/core/dev.c
```

```
+++ b/net/core/dev.c
```

```
@@ -249,15 +249,9 @@ static RAW_NOTIFIER_HEAD(netdev_chain);
```

```
DEFINE_PER_CPU(struct softnet_data, softnet_data);
```

```
-#ifdef CONFIG_SYSFS
```

```
-extern int netdev_sysfs_init(void);
```

```
-extern int netdev_register_sysfs(struct net_device *);
```

```
-extern void netdev_unregister_sysfs(struct net_device *);
```

```

-#else
-#define netdev_sysfs_init() (0)
-#define netdev_register_sysfs(dev) (0)
-#define netdev_unregister_sysfs(dev) do { } while(0)
-#endif
+extern int netdev_kobject_init(void);
+extern int netdev_register_kobject(struct net_device *);
+extern void netdev_unregister_kobject(struct net_device *);

#ifdef CONFIG_DEBUG_LOCK_ALLOC
/*
@@ -3573,7 +3567,7 @@ int register_netdevice(struct net_device *dev)
    if (!dev->rebuild_header)
        dev->rebuild_header = default_rebuild_header;

- ret = netdev_register_sysfs(dev);
+ ret = netdev_register_kobject(dev);
    if (ret)
        goto err_uninit;
    dev->reg_state = NETREG_REGISTERED;
@@ -3836,7 +3830,6 @@ EXPORT_SYMBOL(alloc_netdev_mq);
*/
void free_netdev(struct net_device *dev)
{
-#ifdef CONFIG_SYSFS
    /* Compatibility with error handling in drivers */
    if (dev->reg_state == NETREG_UNINITIALIZED) {
        kfree((char *)dev - dev->padded);
@@ -3848,9 +3841,6 @@ void free_netdev(struct net_device *dev)

    /* will free via device release */
    put_device(&dev->dev);
-#else
- kfree((char *)dev - dev->padded);
-#endif
}

/* Synchronize with packet receive processing. */
@@ -3919,8 +3909,8 @@ void unregister_netdevice(struct net_device *dev)
    /* Notifier chain MUST detach us from master device. */
    BUG_TRAP(!dev->master);

- /* Remove entries from sysfs */
- netdev_unregister_sysfs(dev);
+ /* Remove entries from kobject tree */
+ netdev_unregister_kobject(dev);

    /* Finish processing unregister after unlock */

```

```

    net_set_todo(dev);
@@ -4051,9 +4041,9 @@ int dev_change_net_namespace(struct net_device *dev, struct net
*net, const char
    dev->iflink = dev->ifindex;
}

- /* Fixup sysfs */
+ /* Fixup kobjects */
    err = device_rename(&dev->dev, dev->name);
- BUG_ON(err);
+ WARN_ON(err);

    /* Add the device back in the hashes */
    list_netdevice(dev);
@@ -4356,7 +4346,7 @@ static int __init net_dev_init(void)
    if (dev_proc_init())
        goto out;

- if (netdev_sysfs_init())
+ if (netdev_kobject_init())
    goto out;

    INIT_LIST_HEAD(&ptype_all);
diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
index 79159db..909a03d 100644
--- a/net/core/net-sysfs.c
+++ b/net/core/net-sysfs.c
@@ -18,6 +18,7 @@
#include <linux/wireless.h>
#include <net/iw_handler.h>

+#ifdef CONFIG_SYSFS
static const char fmt_hex[] = "%#x\n";
static const char fmt_long_hex[] = "%#lx\n";
static const char fmt_dec[] = "%d\n";
@@ -392,6 +393,8 @@ static struct attribute_group wireless_group = {
};
#endif

+#endif /* CONFIG_SYSFS */
+
#ifdef CONFIG_HOTPLUG
static int netdev_uevent(struct device *d, char **envp,
    int num_envp, char *buf, int size)
@@ -435,7 +438,9 @@ static void netdev_release(struct device *d)
static struct class net_class = {
    .name = "net",
    .dev_release = netdev_release,

```

```

+msgid CONFIG_SYSFS
    .dev_attrs = net_class_attributes,
+msgid /* CONFIG_SYSFS */
    #ifdef CONFIG_HOTPLUG
        .dev_uevent = netdev_uevent,
    #endif
@@ -444,7 +449,7 @@ static struct class net_class = {
/* Delete sysfs entries but hold kobject reference until after all
 * netdev references are gone.
 */
-void netdev_unregister_sysfs(struct net_device * net)
+void netdev_unregister_kobject(struct net_device * net)
{
    struct device *dev = &(net->dev);

@@ -453,7 +458,7 @@ void netdev_unregister_sysfs(struct net_device * net)
}

/* Create sysfs entries for network device. */
-int netdev_register_sysfs(struct net_device *net)
+int netdev_register_kobject(struct net_device *net)
{
    struct device *dev = &(net->dev);
    struct attribute_group **groups = net->sysfs_groups;
@@ -466,6 +471,7 @@ int netdev_register_sysfs(struct net_device *net)
    BUILD_BUG_ON(BUS_ID_SIZE < IFNAMSIZ);
    strncpy(dev->bus_id, net->name, BUS_ID_SIZE);

+msgid CONFIG_SYSFS
    if (net->get_stats)
        *groups++ = &netstat_group;

@@ -473,11 +479,12 @@ int netdev_register_sysfs(struct net_device *net)
    if (net->wireless_handlers && net->wireless_handlers->get_wireless_stats)
        *groups++ = &wireless_group;
    #endif
+msgid /* CONFIG_SYSFS */

    return device_add(dev);
}

-int netdev_sysfs_init(void)
+int netdev_kobject_init(void)
{
    return class_register(&net_class);
}
--
1.5.3.rc6.17.g1911

```

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

Subject: Re: [PATCH] net: Fix running without sysfs
Posted by [Stephen Hemminger](#) on Thu, 27 Sep 2007 00:07:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 26 Sep 2007 17:48:10 -0600
ebiederm@xmission.com (Eric W. Biederman) wrote:

>
> When sysfs support is compiled out the kernel still keeps and maintains
> the kobject tree. So it is not safe to skip our kobject reference counting or
> to avoid becoming members of the kobject tree. It is safe to not add
> the networking specific sysfs attributes.
>
> This patch removes the sysfs special cases from net/core/dev.c
> renames functions from netdev_sysfs_xxxx to netdev_kobject_xxxx
> and always compiles in net-sysfs.c
>
> net-sysfs.c is modified with a CONFIG_SYSFS guard around the parts
> that are actually sysfs specific.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Acked-by: Stephen Hemminger <shemminger@linux-foundation.org>

Looks good. It also fixes a use after-free in netdev_run_todo
when !CONFIG_SYSFS

--
Stephen Hemminger <shemminger@linux-foundation.org>

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

Subject: Re: [PATCH] net: Fix running without sysfs
Posted by [davem](#) on Thu, 27 Sep 2007 05:03:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: ebiederm@xmission.com (Eric W. Biederman)

Date: Wed, 26 Sep 2007 17:48:10 -0600

>
> When sysfs support is compiled out the kernel still keeps and maintains
> the kobject tree. So it is not safe to skip our kobject reference counting or
> to avoid becoming members of the kobject tree. It is safe to not add
> the networking specific sysfs attributes.
>
> This patch removes the sysfs special cases from net/core/dev.c
> renames functions from netdev_sysfs_xxxx to netdev_kobject_xxxx
> and always compiles in net-sysfs.c
>
> net-sysfs.c is modified with a CONFIG_SYSFS guard around the parts
> that are actually sysfs specific.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Applied to net-2.6.24

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