

---

Subject: [PATCH net-2.6.25 3/3][IPV6] Use ctl paths to register addrconf sysctls  
Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 18:56:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

This looks very much like the patch for ipv4's devinet.

This is also intended to help us with the net namespaces  
and saves the ipv6.ko size by ~320 bytes.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

---

```
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 8b93593..37e2cc0 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -3848,10 +3848,7 @@ static struct addrconf_sysctl_table
{
    struct ctl_table_header *sysctl_header;
    ctl_table addrconf_vars[__NET_IPV6_MAX];
-   ctl_table addrconf_dev[2];
-   ctl_table addrconf_conf_dir[2];
-   ctl_table addrconf_proto_dir[2];
-   ctl_table addrconf_root_dir[2];
+   char *dev_name;
} addrconf_sysctl_read_mostly = {
    .sysctl_header = NULL,
    .addrconf_vars = {
@@ -4072,50 +4069,6 @@ static struct addrconf_sysctl_table
    .ctl_name = 0, /* sentinel */
    },
},
-   .addrconf_dev = {
-   {
-       .ctl_name = NET_PROTO_CONF_ALL,
-       .procname = "all",
-       .mode = 0555,
-       .child = addrconf_sysctl.addrconf_vars,
-   },
-   {
-       .ctl_name = 0, /* sentinel */
-   },
-   },
-   .addrconf_conf_dir = {
-   {
-       .ctl_name = NET_IPV6_CONF,
-       .procname = "conf",
```

```

- .mode = 0555,
- .child = addrconf_sysctl.addrconf_dev,
- },
- {
- .ctl_name = 0, /* sentinel */
- }
- },
- .addrconf_proto_dir = {
- {
- .ctl_name = NET_IPV6,
- .procname = "ipv6",
- .mode = 0555,
- .child = addrconf_sysctl.addrconf_conf_dir,
- },
- {
- .ctl_name = 0, /* sentinel */
- }
- },
- .addrconf_root_dir = {
- {
- .ctl_name = CTL_NET,
- .procname = "net",
- .mode = 0555,
- .child = addrconf_sysctl.addrconf_proto_dir,
- },
- {
- .ctl_name = 0, /* sentinel */
- }
- },
- },
};

```

```

static void __addrconf_sysctl_register(char *dev_name, int ctl_name,
@@ -4124,6 +4077,17 @@ static void __addrconf_sysctl_register(char *dev_name, int ctl_name,
int i;
struct addrconf_sysctl_table *t;

#define ADDRCONF_CTL_PATH_DEV 3
+
+ struct ctl_path addrconf_ctl_path[] = {
+ { .procname = "net", .ctl_name = CTL_NET, },
+ { .procname = "ipv6", .ctl_name = NET_IPV6, },
+ { .procname = "conf", .ctl_name = NET_IPV6_CONF, },
+ { /* to be set */ },
+ { },
+ };
+
+
t = kmemdup(&addrconf_sysctl, sizeof(*t), GFP_KERNEL);

```

```

if (t == NULL)
    goto out;
@@ -4138,19 +4102,15 @@ static void __addrconf_sysctl_register(char *dev_name, int
ctl_name,
    * by sysctl and we wouldn't want anyone to change it under our feet
    * (see SIOCSIFNAME).
    */
- dev_name = kstrdup(dev_name, GFP_KERNEL);
- if (!dev_name)
+ t->dev_name = kstrdup(dev_name, GFP_KERNEL);
+ if (!t->dev_name)
    goto free;

- t->addrconf_dev[0].ctl_name = ctl_name;
- t->addrconf_dev[0].procname = dev_name;
-
- t->addrconf_dev[0].child = t->addrconf_vars;
- t->addrconf_conf_dir[0].child = t->addrconf_dev;
- t->addrconf_proto_dir[0].child = t->addrconf_conf_dir;
- t->addrconf_root_dir[0].child = t->addrconf_proto_dir;
+ addrconf_ctl_path[ADDRCONF_CTL_PATH_DEV].procname = t->dev_name;
+ addrconf_ctl_path[ADDRCONF_CTL_PATH_DEV].ctl_name = ctl_name;

- t->sysctl_header = register_sysctl_table(t->addrconf_root_dir);
+ t->sysctl_header = register_sysctl_paths(addrconf_ctl_path,
+ t->addrconf_vars);
    if (t->sysctl_header == NULL)
        goto free_procname;

@@ -4158,7 +4118,7 @@ static void __addrconf_sysctl_register(char *dev_name, int ctl_name,
    return;

free_procname:
- kfree(dev_name);
+ kfree(t->dev_name);
free:
    kfree(t);
out:
@@ -4178,7 +4138,7 @@ static void addrconf_sysctl_unregister(struct ipv6_devconf *p)
    struct addrconf_sysctl_table *t = p->sysctl;
    p->sysctl = NULL;
    unregister_sysctl_table(t->sysctl_header);
- kfree(t->addrconf_dev[0].procname);
+ kfree(t->dev_name);
    kfree(t);
}
}
--

```

1.5.3.4

---