Subject: [PATCH net-2.6.25 1/19] [NETNS] Add netns parameter to fib_rules_(un)register.
Posted by den on Wed, 09 Jan 2008 17:59:02 GMT
View Forum Message <> Reply to Message

The patch extends the different fib rules API in order to pass the network
namespace pointer. That will allow to access the different tables from
a namespace relative object. As usual, the pointer to the init_net variable
is passed as parameter so we don't break the network.

Acked-by: Benjamin Thery <benjamin.thery@bull.net>
Acked-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/net/fib_rules.h | 4 ++--
 net/core/fib_rules.c | 4 ++--
 net/decnet/dn_rules.c | 4 ++--
 net/ipv4/fib_rules.c | 2 +-
 net/ipv6/fib6_rules.c | 4 ++--
 5 files changed, 9 insertions(+), 9 deletions(-)

```
diff --git a/include/net/fib_rules.h b/include/net/fib_rules.h
index 2364db1..af62345 100644
--- a/include/net/fib_rules.h
+++ b/include/net/fib_rules.h
@@ -101,8 +101,8 @@ static inline u32 frh_get_table(struct fib_rule_hdr *frh, struct nlattr **nla)
  return frh->table;
 }

-extern int   fib_rules_register(struct fib_rules_ops *);
-extern int   fib_rules_unregister(struct fib_rules_ops *);
+extern int fib_rules_register(struct net *, struct fib_rules_ops *);
+extern int fib_rules_unregister(struct net *, struct fib_rules_ops *);
 extern void                    fib_rules_cleanup_ops(struct fib_rules_ops *);

 extern int   fib_rules_lookup(struct fib_rules_ops *,
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
index fcbf41c..ada9c81 100644
--- a/net/core/fib_rules.c
+++ b/net/core/fib_rules.c
@@ -74,7 +74,7 @@ static void flush_route_cache(struct fib_rules_ops *ops)
   ops->flush_cache();
 }

-int fib_rules_register(struct fib_rules_ops *ops)
+int fib_rules_register(struct net *net, struct fib_rules_ops *ops)
 {
  int err = -EEXIST;
```

```
  struct fib_rules_ops *o;
@@ -113,7 +113,7 @@ void fib_rules_cleanup_ops(struct fib_rules_ops *ops)
 }
 EXPORT_SYMBOL_GPL(fib_rules_cleanup_ops);

-int fib_rules_unregister(struct fib_rules_ops *ops)
+int fib_rules_unregister(struct net *net, struct fib_rules_ops *ops)
 {
  int err = 0;
  struct fib_rules_ops *o;
diff --git a/net/decnet/dn_rules.c b/net/decnet/dn_rules.c
index ffebea0..0b5e2b9 100644
--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -255,12 +255,12 @@ void __init dn_fib_rules_init(void)
 {
  BUG_ON(fib_default_rule_add(&dn_fib_rules_ops, 0x7fff,
             RT_TABLE_MAIN, 0));
- fib_rules_register(&dn_fib_rules_ops);
+ fib_rules_register(&init_net, &dn_fib_rules_ops);
 }

 void __exit dn_fib_rules_cleanup(void)
 {
- fib_rules_unregister(&dn_fib_rules_ops);
+ fib_rules_unregister(&init_net, &dn_fib_rules_ops);
 }


diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index a0ada3a..eac3f71 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -314,5 +314,5 @@ static int __init fib_default_rules_init(void)
 void __init fib4_rules_init(void)
 {
  BUG_ON(fib_default_rules_init());
- fib_rules_register(&fib4_rules_ops);
+ fib_rules_register(&init_net, &fib4_rules_ops);
 }
diff --git a/net/ipv6/fib6_rules.c b/net/ipv6/fib6_rules.c
index 9ce2e0a..e4d7e5a 100644
--- a/net/ipv6/fib6_rules.c
+++ b/net/ipv6/fib6_rules.c
@@ -273,7 +273,7 @@ int __init fib6_rules_init(void)
  if (ret)
   goto out;
```

```
- ret = fib_rules_register(&fib6_rules_ops);
+ ret = fib_rules_register(&init_net, &fib6_rules_ops);
  if (ret)
   goto out_default_rules_init;
 out:
@@ -286,5 +286,5 @@ out_default_rules_init:

 void fib6_rules_cleanup(void)
 {
- fib_rules_unregister(&fib6_rules_ops);
+ fib_rules_unregister(&init_net, &fib6_rules_ops);
 }
--
1.5.3.rc5
```