
Subject: [PATCH net-2.6.25 2/19] [NETNS] Pass fib_rules_ops into default_pref method.

Posted by [den](#) on Wed, 19 Dec 2007 15:24:32 GMT

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

fib_rules_ops contains operations and the list of configured rules. ops will become per/namespace soon, so we need them to be known in the default_pref callback.

Acked-by: Benjamin Thery <benjamin.thery@bull.net>

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
include/net/fib_rules.h | 2 +-
net/core/fib_rules.c    | 2 +-
net/decnnet/dn_rules.c  | 2 +-
net/ipv4/fib_rules.c    | 2 +-
net/ipv6/fib6_rules.c   | 2 +-
5 files changed, 5 insertions(+), 5 deletions(-)
```

```
diff --git a/include/net/fib_rules.h b/include/net/fib_rules.h
index af62345..14b8a71 100644
```

```
--- a/include/net/fib_rules.h
+++ b/include/net/fib_rules.h
@@ -56,7 +56,7 @@ struct fib_rules_ops
    int (*fill)(struct fib_rule *, struct sk_buff *,
               struct nlmsg_hdr *,
               struct fib_rule_hdr *);
- u32 (*default_pref)(void);
+ u32 (*default_pref)(struct fib_rules_ops *ops);
    size_t (*nlmsg_payload)(struct fib_rule *);
```

```
/* Called after modifications to the rules set, must flush
```

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

```
index ada9c81..e12e9f5 100644
```

```
--- a/net/core/fib_rules.c
+++ b/net/core/fib_rules.c
@@ -285,7 +285,7 @@ static int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void
*arg)
    rule->table = frh_get_table(frh, tb);

    if (!rule->pref && ops->default_pref)
- rule->pref = ops->default_pref();
+ rule->pref = ops->default_pref(ops);
```

```
    err = -EINVAL;
    if (tb[FRA_GOTO]) {
diff --git a/net/decnnet/dn_rules.c b/net/decnnet/dn_rules.c
index 0b5e2b9..c1fae23 100644
```

```

--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -212,7 +212,7 @@ nla_put_failure:
     return -ENOBUFS;
 }

-static u32 dn_fib_rule_default_pref(void)
+static u32 dn_fib_rule_default_pref(struct fib_rules_ops *ops)
{
    struct list_head *pos;
    struct fib_rule *rule;
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index eac3f71..afe669d 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -245,7 +245,7 @@ nla_put_failure:
     return -ENOBUFS;
 }

-static u32 fib4_rule_default_pref(void)
+static u32 fib4_rule_default_pref(struct fib_rules_ops *ops)
{
    struct list_head *pos;
    struct fib_rule *rule;
diff --git a/net/ipv6/fib6_rules.c b/net/ipv6/fib6_rules.c
index e4d7e5a..76437a1 100644
--- a/net/ipv6/fib6_rules.c
+++ b/net/ipv6/fib6_rules.c
@@ -223,7 +223,7 @@ nla_put_failure:
     return -ENOBUFS;
 }

-static u32 fib6_rule_default_pref(void)
+static u32 fib6_rule_default_pref(struct fib_rules_ops *ops)
{
    return 0x3FFF;
}
--
1.5.3.rc5

```
