

---

Subject: [PATCH net-2.6.25 2/19] [NETNS] Pass fib\_rules\_ops into default\_pref method.

Posted by [den](#) on Wed, 09 Jan 2008 17:59:18 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](mailto:benjamin.thery@bull.net)>

Acked-by: Daniel Lezcano <[dlezcana@fr.ibm.com](mailto:dlezcana@fr.ibm.com)>

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---

```
include/net/fib_rules.h |  2 ++
net/core/fib_rules.c   |  2 ++
net/decnet/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 nlmsghdr *,
           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 nlmsghdr\* 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/decnet/dn\_rules.c b/net/decnet/dn\_rules.c

```

index 0b5e2b9..c1fae23 100644
--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -212,7 +212,7 @@ @ @ -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 @@ @ @ -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 @@ @ @ -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

---