
Subject: [PATCH net-2.6.25 12/19] [NETNS] Show routing information from correct namespace (fib_trie.c)

Posted by [den](#) on Wed, 09 Jan 2008 18:01:07 GMT

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

This is the second part (for the CONFIG_IP_FIB_TRIE case) of the patch #4, where we have created proc files in namespaces.

Now we can dump correct info in them.

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>

```
---
net/ipv4/fib_trie.c | 44 ++++++++++++++++++++++++++++++++++++++-----
1 files changed, 32 insertions(+), 12 deletions(-)

diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index 9526758..708faa9 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -1992,6 +1992,7 @@ struct fib_table *fib_hash_init(u32 id)
#ifdef CONFIG_PROC_FS
/* Depth first Trie walk iterator */
struct fib_trie_iter {
+ struct seq_net_private p;
  struct trie *trie_local, *trie_main;
  struct tnode *tnode;
  struct trie *trie;
@@ -2159,17 +2160,18 @@ static void trie_show_stats(struct seq_file *seq, struct trie_stat *stat)

static int fib_triestat_seq_show(struct seq_file *seq, void *v)
{
+ struct net *net = (struct net *)seq->private;
  struct trie *trie_local, *trie_main;
  struct trie_stat *stat;
  struct fib_table *tb;

  trie_local = NULL;
- tb = fib_get_table(&init_net, RT_TABLE_LOCAL);
+ tb = fib_get_table(net, RT_TABLE_LOCAL);
  if (tb)
    trie_local = (struct trie *) tb->tb_data;

  trie_main = NULL;
- tb = fib_get_table(&init_net, RT_TABLE_MAIN);
+ tb = fib_get_table(net, RT_TABLE_MAIN);
  if (tb)
```

```
trie_main = (struct trie *) tb->tb_data;
```

```
@@ -2199,7 +2201,25 @@ static int fib_triestat_seq_show(struct seq_file *seq, void *v)
```

```
static int fib_triestat_seq_open(struct inode *inode, struct file *file)
```

```
{  
- return single_open(file, fib_triestat_seq_show, NULL);
```

```
+ int err;
```

```
+ struct net *net;
```

```
+
```

```
+ net = get_proc_net(inode);
```

```
+ if (net == NULL)
```

```
+ return -ENXIO;
```

```
+ err = single_open(file, fib_triestat_seq_show, net);
```

```
+ if (err < 0) {
```

```
+ put_net(net);
```

```
+ return err;
```

```
+ }
```

```
+ return 0;
```

```
+
```

```
+
```

```
+static int fib_triestat_seq_release(struct inode *ino, struct file *f)
```

```
+{
```

```
+ struct seq_file *seq = f->private_data;
```

```
+ put_net(seq->private);
```

```
+ return single_release(ino, f);
```

```
}
```

```
static const struct file_operations fib_triestat_fops = {
```

```
@@ -2207,7 +2227,7 @@ static const struct file_operations fib_triestat_fops = {
```

```
.open = fib_triestat_seq_open,
```

```
.read = seq_read,
```

```
.llseek = seq_lseek,
```

```
- .release = single_release,
```

```
+ .release = fib_triestat_seq_release,
```

```
};
```

```
static struct node *fib_trie_get_idx(struct fib_trie_iter *iter,
```

```
@@ -2236,12 +2256,12 @@ static void *fib_trie_seq_start(struct seq_file *seq, loff_t *pos)
```

```
struct fib_table *tb;
```

```
if (!iter->trie_local) {
```

```
- tb = fib_get_table(&init_net, RT_TABLE_LOCAL);
```

```
+ tb = fib_get_table(iter->p.net, RT_TABLE_LOCAL);
```

```
if (tb)
```

```
iter->trie_local = (struct trie *) tb->tb_data;
```

```
}
```

```
if (!iter->trie_main) {
```

```

- tb = fib_get_table(&init_net, RT_TABLE_MAIN);
+ tb = fib_get_table(iter->p.net, RT_TABLE_MAIN);
  if (tb)
    iter->trie_main = (struct trie *) tb->tb_data;
  }
@@ -2385,8 +2405,8 @@ static const struct seq_operations fib_trie_seq_ops = {

static int fib_trie_seq_open(struct inode *inode, struct file *file)
{
- return seq_open_private(file, &fib_trie_seq_ops,
- sizeof(struct fib_trie_iter));
+ return seq_open_net(inode, file, &fib_trie_seq_ops,
+ sizeof(struct fib_trie_iter));
}

static const struct file_operations fib_trie_fops = {
@@ -2394,7 +2414,7 @@ static const struct file_operations fib_trie_fops = {
  .open = fib_trie_seq_open,
  .read = seq_read,
  .llseek = seq_lseek,
- .release = seq_release_private,
+ .release = seq_release_net,
};

static unsigned fib_flag_trans(int type, __be32 mask, const struct fib_info *fi)
@@ -2489,8 +2509,8 @@ static const struct seq_operations fib_route_seq_ops = {

static int fib_route_seq_open(struct inode *inode, struct file *file)
{
- return seq_open_private(file, &fib_route_seq_ops,
- sizeof(struct fib_trie_iter));
+ return seq_open_net(inode, file, &fib_route_seq_ops,
+ sizeof(struct fib_trie_iter));
}

static const struct file_operations fib_route_fops = {
@@ -2498,7 +2518,7 @@ static const struct file_operations fib_route_fops = {
  .open = fib_route_seq_open,
  .read = seq_read,
  .llseek = seq_lseek,
- .release = seq_release_private,
+ .release = seq_release_net,
};

int __net_init fib_proc_init(struct net *net)
--
1.5.3.rc5

```