
Subject: Re: [PATCH] Send quota messages via netlink
Posted by [ebiederm](#) on Wed, 29 Aug 2007 05:41:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Andrew Morton <akpm@linux-foundation.org> writes:

```
> On Tue, 28 Aug 2007 16:13:18 +0200 Jan Kara <jack@suse.cz> wrote:
>
>> Hello,
>>
>> I'm sending rediffed patch implementing sending of quota messages via netlink
>> interface (some rationale in patch description). I've already posted it to
>> LKML some time ago and there were no objections, so I guess it's fine to put
>> it to -mm. Andrew, would you be so kind? Thanks.
>> Userspace daemon reading the messages from the kernel and sending them to
>> dbus and/or user console is also written (it's part of quota-tools). The
>> only remaining problem is there are a few changes needed to libnl needed for
>> the userspace daemon. They were basically acked by the maintainer but it
>> seems he has not merged the patches yet. So this will take a bit more time.
>>
>
> So it's a new kernel->userspace interface.
>
> But we have no description of the interface :(
>
>> +/* Send warning to userspace about user which exceeded quota */
>> +static void send_warning(const struct dquot *dquot, const char warntype)
>> +{
>> + static unsigned long seq;
>> + struct sk_buff *skb;
>> + void *msg_head;
>> + int ret;
>> +
>> + skb = genlmsg_new(QUOTA_NL_MSG_SIZE, GFP_NOFS);
>> + if (!skb) {
>> + printk(KERN_ERR
>> + "VFS: Not enough memory to send quota warning.\n");
>> + return;
>> + }
>> + msg_head = genlmsg_put(skb, 0, seq++, &quota_genl_family, 0,
> QUOTA_NL_C_WARNING);
>> + if (!msg_head) {
>> + printk(KERN_ERR
>> + "VFS: Cannot store netlink header in quota warning.\n");
>> + goto err_out;
>> + }
>> + ret = nla_put_u32(skb, QUOTA_NL_A_QTYPE, dquot->dq_type);
>> + if (ret)
```

```

>> + goto attr_err_out;
>> + ret = nla_put_u64(skb, QUOTA_NL_A_EXCESS_ID, dqot->dq_id);
>> + if (ret)
>> + goto attr_err_out;
>> + ret = nla_put_u32(skb, QUOTA_NL_A_WARNING, warntype);
>> + if (ret)
>> + goto attr_err_out;
>> + ret = nla_put_u32(skb, QUOTA_NL_A_DEV_MAJOR,
>> + MAJOR(dquot->dq_sb->s_dev));
>> + if (ret)
>> + goto attr_err_out;
>> + ret = nla_put_u32(skb, QUOTA_NL_A_DEV_MINOR,
>> + MINOR(dquot->dq_sb->s_dev));
>> + if (ret)
>> + goto attr_err_out;
>> + ret = nla_put_u64(skb, QUOTA_NL_A_CAUSED_ID, current->user->uid);
>> + if (ret)
>> + goto attr_err_out;
>> + genlmsg_end(skb, msg_head);
>> +
>> + ret = genlmsg_multicast(skb, 0, quota_genl_family.id, GFP_NOFS);
>> + if (ret < 0 && ret != -ESRCH)
>> + printk(KERN_ERR
>> + "VFS: Failed to send notification message: %d\n", ret);
>> + return;
>> +attr_err_out:
>> + printk(KERN_ERR "VFS: Failed to compose quota message: %d\n", ret);
>> +err_out:
>> + kfree_skb(skb);
>> +}
>> +#endif
>
> This is it. Normally netlink payloads are represented as a struct. How
> come this one is built-by-hand?

```

No netlink fields (unless I'm confused) are represented as a struct, not the entire netlink payload.

> It doesn't appear to be versioned. Should it be?

Well. If it is using netlink properly each field should have a tag. So it should not need to be versioned, because each field is strictly controlled.

> Does it have (or need) reserved-set-to-zero space for expansion? Again, > hard to tell..

Not if netlink is used properly. Just another nested tag.

> I guess it's OK to send a major and minor out of the kernel like this.
> What's it for? To represent a filesystem? I wonder if there's a more
> modern and useful way of describing the fs. Path to mountpoint or
> something?

Or perhaps the string the fs was mounted with.

> I suspect the namespace virtualisation guys would be interested in a new
> interface which is sending current->user->uid up to userspace. uids are
> per-namespace now. What are the implications? (cc's added)

That we definitely would be. Although the user namespaces is rather strongly incomplete at the moment.

> Is it worth adding a comment explaining why GFP_NOFS is used here?

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
