
Subject: [PATCH] xfs: add newline in debug messages
Posted by [adobriyan](#) on Fri, 29 Dec 2006 15:11:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

cmn_err(CE_DEBUG, ...); doesn't add newline for you.

Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>

```
fs/xfs/quota/xfs_dquot.c    | 32 ++++++-----
fs/xfs/quota/xfs_qm.c       |  8 ++++----
fs/xfs/quota/xfs_qm_syscalls.c | 37 ++++++-----
fs/xfs/quota/xfs_trans_dquot.c |  2 +-
fs/xfs/xfs_log_recover.c    |  2 +-
5 files changed, 41 insertions(+), 40 deletions(-)
```

```
--- a/fs/xfs/quota/xfs_dquot.c
+++ b/fs/xfs/quota/xfs_dquot.c
@@ -923,7 +923,7 @@ #ifdef DEBUG
     if (xfs_do_dqerror) {
         if ((xfs_dqerror_target == mp->m_ddev_targp) &&
             (xfs_dqreq_num++ % xfs_dqerror_mod) == 0) {
-        cmn_err(CE_DEBUG, "Returning error in dqget");
+        cmn_err(CE_DEBUG, "Returning error in dqget\n");
         return (EIO);
     }
 }
@@ -1531,36 +1531,36 @@ #ifdef QUOTADEBUG
void
xfs_qm_dqprint(xfs_dquot_t *dqp)
{
-    cmn_err(CE_DEBUG, "-----KERNEL DQUOT-----");
-    cmn_err(CE_DEBUG, "---- dquotID = %d",
+    cmn_err(CE_DEBUG, "-----KERNEL DQUOT-----\n");
+    cmn_err(CE_DEBUG, "---- dquotID = %d\n",
        (int)be32_to_cpu(dqp->q_core.d_id));
-    cmn_err(CE_DEBUG, "---- type   = %s", DQFLAGTO_TYPESTR(dqp));
-    cmn_err(CE_DEBUG, "---- fs     = 0x%p", dqp->q_mount);
-    cmn_err(CE_DEBUG, "---- blkno  = 0x%x", (int) dqp->q_blkno);
-    cmn_err(CE_DEBUG, "---- boffset = 0x%x", (int) dqp->q_boffset);
-    cmn_err(CE_DEBUG, "---- blkhlimit = %Lu (0x%x)",
+    cmn_err(CE_DEBUG, "---- type   = %s\n", DQFLAGTO_TYPESTR(dqp));
+    cmn_err(CE_DEBUG, "---- fs     = 0x%p\n", dqp->q_mount);
+    cmn_err(CE_DEBUG, "---- blkno  = 0x%x\n", (int) dqp->q_blkno);
+    cmn_err(CE_DEBUG, "---- boffset = 0x%x\n", (int) dqp->q_boffset);
+    cmn_err(CE_DEBUG, "---- blkhlimit = %Lu (0x%x)\n",
        be64_to_cpu(dqp->q_core.d_blk_hardlimit),
        (int)be64_to_cpu(dqp->q_core.d_blk_hardlimit));
```

```

- cmn_err(CE_DEBUG, "---- blkslimit = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "---- blkslimit = %Lu (0x%x)\n",
    be64_to_cpu(dqp->q_core.d_blk_softlimit),
    (int)be64_to_cpu(dqp->q_core.d_blk_softlimit));
- cmn_err(CE_DEBUG, "---- inohlimit = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "---- inohlimit = %Lu (0x%x)\n",
    be64_to_cpu(dqp->q_core.d_ino_hardlimit),
    (int)be64_to_cpu(dqp->q_core.d_ino_hardlimit));
- cmn_err(CE_DEBUG, "---- inoslimit = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "---- inoslimit = %Lu (0x%x)\n",
    be64_to_cpu(dqp->q_core.d_ino_softlimit),
    (int)be64_to_cpu(dqp->q_core.d_ino_softlimit));
- cmn_err(CE_DEBUG, "---- bcount = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "---- bcount = %Lu (0x%x)\n",
    be64_to_cpu(dqp->q_core.d_bcount),
    (int)be64_to_cpu(dqp->q_core.d_bcount));
- cmn_err(CE_DEBUG, "---- icount = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "---- icount = %Lu (0x%x)\n",
    be64_to_cpu(dqp->q_core.d_icount),
    (int)be64_to_cpu(dqp->q_core.d_icount));
- cmn_err(CE_DEBUG, "---- btimer = %d",
+ cmn_err(CE_DEBUG, "---- btimer = %d\n",
    (int)be32_to_cpu(dqp->q_core.d_btimer));
- cmn_err(CE_DEBUG, "---- itimer = %d",
+ cmn_err(CE_DEBUG, "---- itimer = %d\n",
    (int)be32_to_cpu(dqp->q_core.d_itimer));
- cmn_err(CE_DEBUG, "-----");
+ cmn_err(CE_DEBUG, "-----\n");
}
#endif

--- a/fs/xfs/quota/xfs_qm.c
+++ b/fs/xfs/quota/xfs_qm.c
@@ -89,10 +89,10 @@ #ifdef QUOTADEBUG
#define XQM_LIST_PRINT(l, NXT, title) \
{ \
    xfs_dquot_t *dqp; int i = 0; \
- cmn_err(CE_DEBUG, "%s (#%d)", title, (int) (l)->qh_nelems); \
+ cmn_err(CE_DEBUG, "%s (#%d)\n", title, (int) (l)->qh_nelems); \
    for (dqp = (l)->qh_next; dqp != NULL; dqp = dqp->NXT) { \
        cmn_err(CE_DEBUG, "    %d. \"%d (%s)\" " \
-         "bcnt = %d, icnt = %d, refs = %d", \
+         "bcnt = %d, icnt = %d, refs = %d\n", \
            ++i, (int) be32_to_cpu(dqp->q_core.d_id), \
            DQFLAGTO_TYPESTR(dqp), \
            (int) be64_to_cpu(dqp->q_core.d_bcount), \
@@ -2065,7 +2065,7 @@ xfs_qm_shake_freelist(
    nflushes = 0;

```

```

#ifdef QUOTADDEBUG
- cmn_err(CE_DEBUG, "Shake free 0x%x", howmany);
+ cmn_err(CE_DEBUG, "Shake free 0x%x\n", howmany);
#endif
/* lock order is : hashchainlock, freelistlock, mplistlock */
tryagain:
@@ -2781,7 +2781,7 @@ xfs_qm_freelist_destroy(xfs_frlist_t *ql
    xfs_dqlock(dqp);
    nextdqp = dqp->dq_flnext;
#ifdef QUOTADDEBUG
- cmn_err(CE_DEBUG, "FREELIST destroy 0x%p", dqp);
+ cmn_err(CE_DEBUG, "FREELIST destroy 0x%p\n", dqp);
#endif
    XQM_FREELIST_REMOVE(dqp);
    xfs_dqunlock(dqp);
--- a/fs/xfs/quota/xfs_qm_syscalls.c
+++ b/fs/xfs/quota/xfs_qm_syscalls.c
@@ -1145,10 +1145,10 @@ #define DQTEST_HASH(mp, id, type) ((ty
#define DQTEST_LIST_PRINT(l, NXT, title) \
{ \
    xfs_dqtest_t *dqp; int i = 0;\
- cmn_err(CE_DEBUG, "%s (#%d)", title, (int) (l)->qh_nelems); \
+ cmn_err(CE_DEBUG, "%s (#%d)\n", title, (int) (l)->qh_nelems); \
    for (dqp = (xfs_dqtest_t *) (l)->qh_next; dqp != NULL; \
        dqp = (xfs_dqtest_t *) dqp->NXT) { \
- cmn_err(CE_DEBUG, " %d. \"%d (%s)\" bcnt = %d, icnt = %d", \
+ cmn_err(CE_DEBUG, " %d. \"%d (%s)\" bcnt = %d, icnt = %d\n", \
        ++i, dqp->d_id, DQFLAGTO_TYPESTR(dqp), \
        dqp->d_bcount, dqp->d_icount); } \
}
@@ -1178,14 +1178,14 @@ STATIC void
xfs_qm_dqtest_print(
    xfs_dqtest_t *d)
{
- cmn_err(CE_DEBUG, "-----DQTEST DQUOT-----");
- cmn_err(CE_DEBUG, "---- dquot ID = %d", d->d_id);
- cmn_err(CE_DEBUG, "---- fs      = 0x%p", d->q_mount);
- cmn_err(CE_DEBUG, "---- bcount  = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "-----DQTEST DQUOT-----\n");
+ cmn_err(CE_DEBUG, "---- dquot ID = %d\n", d->d_id);
+ cmn_err(CE_DEBUG, "---- fs      = 0x%p\n", d->q_mount);
+ cmn_err(CE_DEBUG, "---- bcount  = %Lu (0x%x)\n",
    d->d_bcount, (int)d->d_bcount);
- cmn_err(CE_DEBUG, "---- icount  = %Lu (0x%x)",
+ cmn_err(CE_DEBUG, "---- icount  = %Lu (0x%x)\n",
    d->d_icount, (int)d->d_icount);
- cmn_err(CE_DEBUG, "-----");

```

```

+ cmn_err(CE_DEBUG, "-----\n");
}

STATIC void
@@ -1199,10 +1199,11 @@ xfs_qm_dqtest_failed(
{
    qmtest_nfails++;
    if (error)
-   cmn_err(CE_DEBUG, "quotacheck failed id=%d, err=%d\nreason: %s",
+   cmn_err(CE_DEBUG,
+   "quotacheck failed id=%d, err=%d, reason: %s\n",
        d->d_id, error, reason);
    else
-   cmn_err(CE_DEBUG, "quotacheck failed id=%d (%s) [%d != %d]",
+   cmn_err(CE_DEBUG, "quotacheck failed id=%d (%s) [%d != %d]\n",
        d->d_id, reason, (int)a, (int)b);
    xfs_qm_dqtest_print(d);
    if (dqp)
@@ -1232,7 +1233,7 @@ xfs_dqtest_cmp2(
    be64_to_cpu(dqp->q_core.d_blk_softlimit)) {
    if (!dqp->q_core.d_btimer && dqp->q_core.d_id) {
        cmn_err(CE_DEBUG,
-        "%d [%s] [0x%p] BLK TIMER NOT STARTED",
+        "%d [%s] [0x%p] BLK TIMER NOT STARTED\n",
        d->d_id, DQFLAGTO_TYPESTR(d), d->q_mount);
        err++;
    }
}
@@ -1242,14 +1243,14 @@ xfs_dqtest_cmp2(
    be64_to_cpu(dqp->q_core.d_ino_softlimit)) {
    if (!dqp->q_core.d_itimer && dqp->q_core.d_id) {
        cmn_err(CE_DEBUG,
-        "%d [%s] [0x%p] INO TIMER NOT STARTED",
+        "%d [%s] [0x%p] INO TIMER NOT STARTED\n",
        d->d_id, DQFLAGTO_TYPESTR(d), d->q_mount);
        err++;
    }
}
}
#ifdef QUOTADDEBUG
    if (!err) {
-   cmn_err(CE_DEBUG, "%d [%s] [0x%p] qchecked",
+   cmn_err(CE_DEBUG, "%d [%s] [0x%p] qchecked\n",
        d->d_id, DQFLAGTO_TYPESTR(d), d->q_mount);
    }
#endif
@@ -1446,9 +1447,9 @@ xfs_qm_internalqcheck(
}
} while (!done);
if (error) {

```

```

- cmn_err(CE_DEBUG, "Bulkstat returned error 0x%x", error);
+ cmn_err(CE_DEBUG, "Bulkstat returned error 0x%x\n", error);
}
- cmn_err(CE_DEBUG, "Checking results against system dquotes");
+ cmn_err(CE_DEBUG, "Checking results against system dquotes\n");
for (i = 0; i < qmtest_hashmask; i++) {
    h1 = &qmtest_udqtab[i];
    for (d = (xfs_dqtest_t *) h1->qh_next; d != NULL; ) {
@@ -1467,10 +1468,10 @@ xfs_qm_internalqcheck(
    }

    if (qmtest_nfails) {
- cmn_err(CE_DEBUG, "***** quotacheck failed *****");
- cmn_err(CE_DEBUG, "failures = %d", qmtest_nfails);
+ cmn_err(CE_DEBUG, "***** quotacheck failed *****\n");
+ cmn_err(CE_DEBUG, "failures = %d\n", qmtest_nfails);
    } else {
- cmn_err(CE_DEBUG, "***** quotacheck successful! *****");
+ cmn_err(CE_DEBUG, "***** quotacheck successful! *****\n");
    }
    kmem_free(qmtest_udqtab, qmtest_hashmask * sizeof(xfs_dqhash_t));
    kmem_free(qmtest_gdqtab, qmtest_hashmask * sizeof(xfs_dqhash_t));
--- a/fs/xfs/quota/xfs_trans_dquot.c
+++ b/fs/xfs/quota/xfs_trans_dquot.c
@@ -661,7 +661,7 @@ xfs_trans_dqresv(
    XFS_IS_QUOTA_ENFORCED(dqp->q_mount)) {
#ifdef QUOTADEBUG
    cmn_err(CE_DEBUG, "BLK Res: nblks=%ld + resbcount=%ld"
-    " > hardlimit=%ld?", nblks, *resbcountp, hardlimit);
+    " > hardlimit=%ld?\n", nblks, *resbcountp, hardlimit);
#endif
    if (nblks > 0) {
/*
--- a/fs/xfs/xfs_log_recover.c
+++ b/fs/xfs/xfs_log_recover.c
@@ -3447,7 +3447,7 @@ xlog_unpack_data_checksum(
    "XFS: LogR chksum mismatch: was (0x%x) is (0x%x)\n",
    INT_GET(rhead->h_chksum, ARCH_CONVERT), chksum);
    cmn_err(CE_DEBUG,
-"XFS: Disregard message if filesystem was created with non-DEBUG kernel");
+"XFS: Disregard message if filesystem was created with non-DEBUG kernel\n");
    if (XFS_SB_VERSION_HASLOGV2(&log->l_mp->m_sb)) {
        cmn_err(CE_DEBUG,
        "XFS: LogR this is a LogV2 filesystem\n");

```

Subject: Re: [xfs-masters] [PATCH] xfs: add newline in debug messages

Posted by [Eric Sandeen](#) on Sat, 30 Dec 2006 18:40:30 GMT

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

Alexey Dobriyan wrote:

> `cmn_err(CE_DEBUG, ...)`; doesn't add newline for you.

The patch looks about right to me. This -is- how it was -intended- to be used.

However, the no-newline behavior is so rarely used that I almost wonder if it'd be more straightforward to drop it, and instead use `sprintf` or something to construct those strings... otherwise I'm sure that future `CE_DEBUG` messages will forget newlines as well.

-Eric
