

```

*****
15876 Thu Jan 17 14:55:18 2019
new/usr/src/lib/fm/libfmevent/common/fmev_subscribe.c
10114 fmev_proxy_cb() doesn't need inval checks
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 */

26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28 */

30 /*
31  * FMA event subscription interfaces - subscribe to FMA protocol
32  * from outside the fault manager.
33 */

35 #include <sys/types.h>
36 #include <atomic.h>
37 #include <libsysevent.h>
38 #include <libuutil.h>
39 #include <pthread.h>
40 #include <stdarg.h>
41 #include <stdlib.h>
42 #include <string.h>
43 #include <strings.h>
44 #include <unistd.h>
45 #include <fm/libtopo.h>

47 #include <fm/libfmevent.h>

49 #include "fmev_impl.h"

51 static topo_hdl_t *g_topohdl;

53 typedef struct {
54     struct fmev_hdl_cmn sh_cmn;
55     evchan_t *sh_binding;
56     uu_avl_pool_t *sh_pool;
57     uu_avl_t *sh_avl;
58     uint32_t sh_subcnt;
59     uint32_t sh_flags;
60     sysevent_subattr_t *sh_attr;
61     pthread_mutex_t sh_lock;

```

```

62     pthread_mutex_t sh_srlz_lock;
63 } fmev_shdl_impl_t;
_____unchanged_portion_omitted_____

207 /*
208  * Our door service function. We return 0 regardless so that the kernel
209  * does not keep either retrying (EAGAIN) or bleat to cmn_err.
210 */

208 uint64_t fmev_proxy_cb_inval;
212 uint64_t fmev_proxy_cb_enomem;

214 int
215 fmev_proxy_cb(sysevent_t *sep, void *arg)
216 {
217     struct fmev_subinfo *sip = arg;
218     fmev_shdl_impl_t *ihdl = sip->si_ihdl;
219     nvlist_t *nvl;
220     char *class;
221     fmev_t ev;

220     if (sip == NULL || sip->si_cb == NULL) {
221         fmev_proxy_cb_inval++;
222         return (0);
223     }

223     if ((ev = fmev_sysev2fmev(IHDL2HDL(ihdl), sep, &class, &nvl)) == NULL) {
224         fmev_proxy_cb_enomem++;
225         return (0);
226     }

228     if (ihdl->sh_flags & SHDL_FL_SERIALIZE)
229         (void) pthread_mutex_lock(&ihdl->sh_srlz_lock);

231     sip->si_cb(ev, class, nvl, sip->si_cbag);

233     if (ihdl->sh_flags & SHDL_FL_SERIALIZE)
234         (void) pthread_mutex_unlock(&ihdl->sh_srlz_lock);

236     fmev_rele(ev); /* release hold obtained in fmev_sysev2fmev */

238     return (0);
239 }
_____unchanged_portion_omitted_____

```