```
**********************************************************
   20533 Thu Jan 17 14:53:10 2019
new/usr/src/cmd/fm/fmstat/common/fmstat.c
10113 fmd_adm_xprt_f should return void
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright (c) 2018, Joyent, Inc.
 */

#include <fm/fmd_adm.h>

#include <strings.h>
#include <limits.h>
#include <stdlib.h>
#include <stdarg.h>
#include <stdio.h>
#include <errno.h>
#include <poll.h>
#include <locale.h>

#include "statcommon.h"

#define FMSTAT_EXIT_SUCCESS     0
#define FMSTAT_EXIT_ERROR       1
#define FMSTAT_EXIT_USAGE       2

static const struct stats {
        fmd_stat_t module;
        fmd_stat_t authority;
        fmd_stat_t state;
        fmd_stat_t loadtime;
        fmd_stat_t snaptime;
        fmd_stat_t received;
        fmd_stat_t discarded;
        fmd_stat_t retried;
        fmd_stat_t replayed;
        fmd_stat_t lost;
        fmd_stat_t dispatched;
        fmd_stat_t dequeued;
        fmd_stat_t prdequeued;
```

```
        fmd_stat_t accepted;
        fmd_stat_t memtotal;
        fmd_stat_t buftotal;
        fmd_stat_t caseopen;
        fmd_stat_t casesolved;
        fmd_stat_t wcnt;
        fmd_stat_t wtime;
        fmd_stat_t wlentime;
        fmd_stat_t wlastupdate;
        fmd_stat_t dtime;
        fmd_stat_t dlastupdate;
} stats_template = {
______unchanged_portion_omitted_

/*ARGSUSED*/
static void
static int
stat_one_xprt(id_t id, void *ignored)
{
        fmd_adm_stats_t ams;
        struct modstats *mp;

        if (fmd_adm_xprt_stats(g_adm, id, &ams) != 0) {
                warn("failed to retrieve statistics for transport %d", (int)id);
                return;
                return (0); /* continue on to the next transport */
        }

        for (mp = g_mods; mp != NULL; mp = mp->m_next) {
                if (mp->m_id == id)
                        break;
        }

        if (mp == NULL && (mp = modstat_create(NULL, id)) == NULL) {
                warn("failed to allocate memory for transport %d", (int)id);
                (void) fmd_adm_stats_free(g_adm, &ams);
                return;
                return (0);
        }

        modstat_compute(mp, &ams);

        (void) printf("%3d %5s %7llu %7llu %7llu %7llu "
            "%4.1f %6.1f %3.0f %3.0f %s\n", (int)id,
            mp->m_new->state.fmds_value.str,
            u64delta(mp->m_old->prdequeued.fmds_value.ui64,
            mp->m_new->prdequeued.fmds_value.ui64),
            u64delta(mp->m_old->received.fmds_value.ui64,
            mp->m_new->received.fmds_value.ui64),
            u64delta(mp->m_old->discarded.fmds_value.ui64,
            mp->m_new->discarded.fmds_value.ui64),
            u64delta(mp->m_old->lost.fmds_value.ui64,
            mp->m_new->lost.fmds_value.ui64),
            mp->m_wait, mp->m_svc, mp->m_pct_w, mp->m_pct_b,
            mp->m_new->module.fmds_value.str);

        (void) fmd_adm_stats_free(g_adm, &ams);
        return (0);
}
______unchanged_portion_omitted_

static void
static int
stat_one_xprt_auth(id_t id, void *arg)
{
        const char *module = arg;
        fmd_adm_stats_t ams;
        struct modstats *mp;

        if (fmd_adm_xprt_stats(g_adm, id, &ams) != 0) {
                warn("failed to retrieve statistics for transport %d", (int)id);
                return;
                return (0); /* continue on to the next transport */
        }

        for (mp = g_mods; mp != NULL; mp = mp->m_next) {
                if (mp->m_id == id)
                        break;
        }

        if (mp == NULL && (mp = modstat_create(NULL, id)) == NULL) {
                warn("failed to allocate memory for transport %d", (int)id);
                (void) fmd_adm_stats_free(g_adm, &ams);
                return;
                return (0);
        }

        modstat_compute(mp, &ams);

        if (module == NULL ||
            strcmp(module, mp->m_new->module.fmds_value.str) == 0) {
                (void) printf("%3d %5s %-18s  %s\n", (int)id,
                    mp->m_new->state.fmds_value.str,
                    mp->m_new->module.fmds_value.str,
                    mp->m_new->authority.fmds_value.str ?
                    mp->m_new->authority.fmds_value.str : "-");
        }

        (void) fmd_adm_stats_free(g_adm, &ams);
        return (0);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    5147 Thu Jan 17 14:53:10 2019
new/usr/src/lib/fm/libfmd_adm/common/fmd_adm.h
10113 fmd_adm_xprt_f should return void
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright (c) 2018, Joyent, Inc.
 */

#ifndef _FMD_ADM_H
#define _FMD_ADM_H

#pragma ident   "%Z%%M% %I%     %E% SMI"

#include <fm/fmd_api.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * Fault Management Daemon Administrative Interfaces
 *
 * Note: The contents of this file are private to the implementation of the
 * Solaris system and FMD subsystem and are subject to change at any time
 * without notice.  Applications and drivers using these interfaces will fail
 * to run on future releases.  These interfaces should not be used for any
 * purpose until they are publicly documented for use outside of Sun.
 */

#define FMD_ADM_VERSION 1               /* library ABI interface version */
#define FMD_ADM_PROGRAM 0               /* connect library to system fmd */

typedef struct fmd_adm fmd_adm_t;

extern fmd_adm_t *fmd_adm_open(const char *, uint32_t, int);
extern void fmd_adm_close(fmd_adm_t *);
extern const char *fmd_adm_errmsg(fmd_adm_t *);

typedef struct fmd_adm_stats {
```

```
        fmd_stat_t *ams_buf;            /* statistics data array */
        uint_t ams_len;                 /* length of data array */
} fmd_adm_stats_t;
______unchanged_portion_omitted_

#define FMD_ADM_SERD_FIRED      0x1     /* serd engine has fired */

typedef int fmd_adm_serd_f(const fmd_adm_serdinfo_t *, void *);

extern int fmd_adm_serd_iter(fmd_adm_t *, const char *,
    fmd_adm_serd_f *, void *);
extern int fmd_adm_serd_reset(fmd_adm_t *, const char *, const char *);

typedef void fmd_adm_xprt_f(id_t, void *);
typedef int fmd_adm_xprt_f(id_t, void *);

extern int fmd_adm_xprt_iter(fmd_adm_t *, fmd_adm_xprt_f *, void *);
extern int fmd_adm_xprt_stats(fmd_adm_t *, id_t, fmd_adm_stats_t *);

extern int fmd_adm_log_rotate(fmd_adm_t *, const char *);

#ifdef  __cplusplus
}
______unchanged_portion_omitted_
```