```
**********************************************************
  200633 Wed Jul  9 12:27:24 2014
new/usr/src/lib/libzonecfg/common/libzonecfg.c
3347 zonecfg(1M) is confused about selection
**********************************************************
/*
 * 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 2014 Gary Mills
 * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
 */

#include <libsysevent.h>
#include <pthread.h>
#include <stdlib.h>
#include <errno.h>
#include <fnmatch.h>
#include <strings.h>
#include <unistd.h>
#include <assert.h>
#include <libgen.h>
#include <libintl.h>
#include <alloca.h>
#include <ctype.h>
#include <sys/acl.h>
#include <sys/stat.h>
#include <sys/brand.h>
#include <sys/mntio.h>
#include <sys/mnttab.h>
#include <sys/nvpair.h>
#include <sys/types.h>
#include <sys/sockio.h>
#include <sys/systeminfo.h>
#include <ftw.h>
#include <pool.h>
#include <libscf.h>
#include <libproc.h>
#include <sys/priocntl.h>
#include <libuutil.h>
#include <wait.h>
#include <bsm/adt.h>
#include <auth_attr.h>
#include <auth_list.h>
#include <secdb.h>
#include <user_attr.h>
#include <prof_attr.h>
```



```
#include <arpa/inet.h>
#include <netdb.h>

#include <libxml/xmlmemory.h>
#include <libxml/parser.h>

#include <libdevinfo.h>
#include <uuid/uuid.h>
#include <dirent.h>
#include <libbrand.h>

#include <libzonecfg.h>
#include "zonecfg_impl.h"

#define _PATH_TMPFILE   "/zonecfg.XXXXXX"
#define ZONE_CB_RETRY_COUNT             10
#define ZONE_EVENT_PING_SUBCLASS        "ping"
#define ZONE_EVENT_PING_PUBLISHER       "solaris"

/* Hard-code the DTD element/attribute/entity names just once, here. */
#define DTD_ELEM_ATTR           (const xmlChar *) "attr"
#define DTD_ELEM_COMMENT        (const xmlChar *) "comment"
#define DTD_ELEM_DEVICE         (const xmlChar *) "device"
#define DTD_ELEM_FS             (const xmlChar *) "filesystem"
#define DTD_ELEM_FSOPTION       (const xmlChar *) "fsoption"
#define DTD_ELEM_NET            (const xmlChar *) "network"
#define DTD_ELEM_RCTL           (const xmlChar *) "rctl"
#define DTD_ELEM_RCTLVALUE      (const xmlChar *) "rctl-value"
#define DTD_ELEM_ZONE           (const xmlChar *) "zone"
#define DTD_ELEM_DATASET        (const xmlChar *) "dataset"
#define DTD_ELEM_TMPPOOL        (const xmlChar *) "tmp_pool"
#define DTD_ELEM_PSET           (const xmlChar *) "pset"
#define DTD_ELEM_MCAP           (const xmlChar *) "mcap"
#define DTD_ELEM_PACKAGE        (const xmlChar *) "package"
#define DTD_ELEM_OBSOLETES      (const xmlChar *) "obsoletes"
#define DTD_ELEM_DEV_PERM       (const xmlChar *) "dev-perm"
#define DTD_ELEM_ADMIN          (const xmlChar *) "admin"

#define DTD_ATTR_ACTION         (const xmlChar *) "action"
#define DTD_ATTR_ADDRESS        (const xmlChar *) "address"
#define DTD_ATTR_ALLOWED_ADDRESS        (const xmlChar *) "allowed-address"
#define DTD_ATTR_AUTOBOOT       (const xmlChar *) "autoboot"
#define DTD_ATTR_IPTYPE         (const xmlChar *) "ip-type"
#define DTD_ATTR_DEFROUTER      (const xmlChar *) "defrouter"
#define DTD_ATTR_DIR            (const xmlChar *) "directory"
#define DTD_ATTR_LIMIT          (const xmlChar *) "limit"
#define DTD_ATTR_LIMITPRIV      (const xmlChar *) "limitpriv"
#define DTD_ATTR_BOOTARGS       (const xmlChar *) "bootargs"
#define DTD_ATTR_SCHED          (const xmlChar *) "scheduling-class"
#define DTD_ATTR_MATCH          (const xmlChar *) "match"
#define DTD_ATTR_NAME           (const xmlChar *) "name"
#define DTD_ATTR_PHYSICAL       (const xmlChar *) "physical"
#define DTD_ATTR_POOL           (const xmlChar *) "pool"
#define DTD_ATTR_PRIV           (const xmlChar *) "priv"
#define DTD_ATTR_RAW            (const xmlChar *) "raw"
#define DTD_ATTR_SPECIAL        (const xmlChar *) "special"
#define DTD_ATTR_TYPE           (const xmlChar *) "type"
#define DTD_ATTR_VALUE          (const xmlChar *) "value"
#define DTD_ATTR_ZONEPATH       (const xmlChar *) "zonepath"
#define DTD_ATTR_NCPU_MIN       (const xmlChar *) "ncpu_min"
#define DTD_ATTR_NCPU_MAX       (const xmlChar *) "ncpu_max"
#define DTD_ATTR_IMPORTANCE     (const xmlChar *) "importance"
#define DTD_ATTR_PHYSCAP        (const xmlChar *) "physcap"
#define DTD_ATTR_VERSION        (const xmlChar *) "version"
#define DTD_ATTR_ID             (const xmlChar *) "id"
#define DTD_ATTR_UID            (const xmlChar *) "uid"
```

```
#define DTD_ATTR_GID            (const xmlChar *) "gid"
#define DTD_ATTR_MODE           (const xmlChar *) "mode"
#define DTD_ATTR_ACL            (const xmlChar *) "acl"
#define DTD_ATTR_BRAND          (const xmlChar *) "brand"
#define DTD_ATTR_HOSTID         (const xmlChar *) "hostid"
#define DTD_ATTR_USER           (const xmlChar *) "user"
#define DTD_ATTR_AUTHS          (const xmlChar *) "auths"
#define DTD_ATTR_FS_ALLOWED     (const xmlChar *) "fs-allowed"

#define DTD_ENTITY_BOOLEAN      "boolean"
#define DTD_ENTITY_DEVPATH      "devpath"
#define DTD_ENTITY_DRIVER       "driver"
#define DTD_ENTITY_DRVMIN       "drv_min"
#define DTD_ENTITY_FALSE        "false"
#define DTD_ENTITY_INT          "int"
#define DTD_ENTITY_STRING       "string"
#define DTD_ENTITY_TRUE         "true"
#define DTD_ENTITY_UINT         "uint"

#define DTD_ENTITY_BOOL_LEN     6       /* "false" */

#define ATTACH_FORCED   "SUNWattached.xml"

#define TMP_POOL_NAME   "SUNWtmp_%s"
#define MAX_TMP_POOL_NAME       (ZONENAME_MAX + 9)
#define RCAP_SERVICE    "system/rcap:default"
#define POOLD_SERVICE   "system/pools/dynamic:default"

/*
 * rctl alias definitions
 *
 * This holds the alias, the full rctl name, the default priv value, action
 * and lower limit.  The functions that handle rctl aliases step through
 * this table, matching on the alias, and using the full values for setting
 * the rctl entry as well the limit for validation.
 */
static struct alias {
        char *shortname;
        char *realname;
        char *priv;
        char *action;
        uint64_t low_limit;
} aliases[] = {
```

unchanged_portion_omitted_

```
static boolean_t
match_prop(xmlNodePtr cur, const xmlChar *attr, char *user_prop)
{
        xmlChar *gotten_prop;
        int prop_result;

        gotten_prop = xmlGetProp(cur, attr);
        if (gotten_prop == NULL)        /* shouldn't happen */
                return (B_FALSE);
        prop_result = xmlStrcmp(gotten_prop, (const xmlChar *) user_prop);
        xmlFree(gotten_prop);
        return ((prop_result == 0));    /* empty strings will match */
        return ((prop_result == 0));
}
```

unchanged_portion_omitted_

```
static int
zonecfg_delete_nwif_core(zone_dochandle_t handle, struct zone_nwiftab *tabptr)
{
        xmlNodePtr cur = handle->zone_dh_cur;
        boolean_t addr_match, phys_match, allowed_addr_match;
```



```
        for (cur = cur->xmlChildrenNode; cur != NULL; cur = cur->next) {
                if (xmlStrcmp(cur->name, DTD_ELEM_NET))
                        continue;

                addr_match = match_prop(cur, DTD_ATTR_ADDRESS,
                    tabptr->zone_nwif_address);
                allowed_addr_match = match_prop(cur, DTD_ATTR_ALLOWED_ADDRESS,
                    tabptr->zone_nwif_allowed_address);
                phys_match = match_prop(cur, DTD_ATTR_PHYSICAL,
                    tabptr->zone_nwif_physical);

                if (addr_match && allowed_addr_match && phys_match) {
                if ((addr_match || allowed_addr_match) && phys_match) {
                        xmlUnlinkNode(cur);
                        xmlFreeNode(cur);
                        return (Z_OK);
                }
        }
        return (Z_NO_RESOURCE_ID);
}
```

unchanged_portion_omitted_