```
**********************************************************
   77394 Thu Aug  1 18:13:05 2013
new/usr/src/cmd/mdb/common/mdb/mdb_print.c
3953 Calling ::list without specifying the name of the next member causes mdb to
Reviewed by: Christopher Siden <christopher.siden@delphix.com>
Reviewed by: Matthew Ahrens <mahrens@delphix.com>
**********************************************************
```

```
/*
 * 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) 2013 by Delphix. All rights reserved.
 * Copyright (c) 2012 by Delphix. All rights reserved.
 * Copyright (c) 2012 Joyent, Inc. All rights reserved.
 */

#include <mdb/mdb_modapi.h>
#include <mdb/mdb_target.h>
#include <mdb/mdb_argvec.h>
#include <mdb/mdb_string.h>
#include <mdb/mdb_stdlib.h>
#include <mdb/mdb_err.h>
#include <mdb/mdb_debug.h>
#include <mdb/mdb_fmt.h>
#include <mdb/mdb_ctf.h>
#include <mdb/mdb_ctf_impl.h>
#include <mdb/mdb.h>
#include <mdb/mdb_tab.h>

#include <sys/isa_defs.h>
#include <sys/param.h>
#include <sys/sysmacros.h>
#include <netinet/in.h>
#include <strings.h>
#include <libctf.h>
#include <ctype.h>

typedef struct holeinfo {
        ulong_t hi_offset;              /* expected offset */
        uchar_t hi_isunion;             /* represents a union */
} holeinfo_t;
```

*unchanged_portion_omitted_*

```
/*ARGSUSED*/
```



```
int
cmd_list(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
{
        int offset;
        uintptr_t a, tmp;
        int ret;

        if (!(flags & DCMD_ADDRSPEC) || argc == 0)
                return (DCMD_USAGE);

        if (argv->a_type != MDB_TYPE_STRING) {
                /*
                 * We are being given a raw offset in lieu of a type and
                 * member; confirm the number of arguments and argument
                 * type.
                 * member; confirm the arguments.
                 */
                if (argc != 1 || argv->a_type != MDB_TYPE_IMMEDIATE)
                if (argv->a_type != MDB_TYPE_IMMEDIATE)
                        return (DCMD_USAGE);

                offset = argv->a_un.a_val;

                argv++;
                argc--;

                if (offset % sizeof (uintptr_t)) {
                        mdb_warn("offset must fall on a word boundary\n");
                        return (DCMD_ABORT);
                }
        } else {
                const char *member;
                char buf[MDB_SYM_NAMLEN];
                int ret;

                /*
                 * Check that we were provided 2 arguments: a type name
                 * and a member of that type.
                 */
                if (argc != 2)
                        return (DCMD_USAGE);

                ret = args_to_typename(&argc, &argv, buf, sizeof (buf));
                if (ret != 0)
                        return (ret);

                argv++;
                argc--;

                member = argv->a_un.a_str;
                offset = mdb_ctf_offsetof_by_name(buf, member);
                if (offset == -1)
                        return (DCMD_ABORT);

                argv++;
                argc--;

                if (offset % (sizeof (uintptr_t)) != 0) {
                        mdb_warn("%s is not a word-aligned member\n", member);
                        return (DCMD_ABORT);
                }
        }

        /*
         * If we have any unchewed arguments, a variable name must be present.
         */
```

```
761         if (argc == 1) {
762                 if (argv->a_type != MDB_TYPE_STRING)
763                         return (DCMD_USAGE);

765                 if ((ret = setup_vcb(argv->a_un.a_str, addr)) != 0)
766                         return (ret);

768         } else if (argc != 0) {
769                 return (DCMD_USAGE);
770         }

772         a = addr;

774         do {
775                 mdb_printf("%lr\n", a);

777                 if (mdb_vread(&tmp, sizeof (tmp), a + offset) == -1) {
778                         mdb_warn("failed to read next pointer from object %p",
779                             a);
780                         return (DCMD_ERR);
781                 }

783                 a = tmp;
784         } while (a != addr && a != NULL);

786         return (DCMD_OK);
787 }
```

unchanged_portion_omitted_