

new/usr/src/cmd/printf/printf.c

1

```
*****
12863 Fri May 9 12:12:12 2014
new/usr/src/cmd/printf/printf.c
4854 printf(1) doesn't support %b and \c properly
4818 printf(1) should support n$ width and precision specifiers
*****
1 /*
2 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
3 * Copyright 2010 Nexenta Systems, Inc. All rights reserved.
4 * Copyright (c) 1989, 1993
5 *      The Regents of the University of California. All rights reserved.
6 *
7 * Redistribution and use in source and binary forms, with or without
8 * modification, are permitted provided that the following conditions
9 * are met:
10 * 1. Redistributions of source code must retain the above copyright
11 *    notice, this list of conditions and the following disclaimer.
12 * 2. Redistributions in binary form must reproduce the above copyright
13 *    notice, this list of conditions and the following disclaimer in the
14 *    documentation and/or other materials provided with the distribution.
15 * 4. Neither the name of the University nor the names of its contributors
16 *    may be used to endorse or promote products derived from this software
17 *    without specific prior written permission.
18 *
19 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
20 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
21 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
22 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
23 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
24 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
25 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
26 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
27 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
28 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
29 * SUCH DAMAGE.
30 */

32 #include <sys/types.h>

34 #include <err.h>
35 #include <errno.h>
36 #include <inttypes.h>
37 #include <limits.h>
38 #include <stdio.h>
39 #include <stdlib.h>
40 #include <string.h>
41 #include <unistd.h>
42 #include <alloca.h>
43 #include <ctype.h>
44 #include <locale.h>
45 #include <note.h>

47 #define warnx1(a, b, c)      warnx(a)
48 #define warnx2(a, b, c)      warnx(a, b)
49 #define warnx3(a, b, c)      warnx(a, b, c)

51 #define PTRDIFF(x, y)  ((uintptr_t)(x) - (uintptr_t)(y))

53 #define _(x)    gettext(x)

55 #define PF(f, func) do {
56     char *b = NULL;
57     int dollar = 0;
58     if (*f == '$') {
59         dollar++;
60         *f = '%';
61     }
62     if (havewidth)
63         if (haveprec)
64             (void) asprintf(&b, f, fieldwidth, precision, func);
65         else
66             (void) asprintf(&b, f, fieldwidth, func);
67     else if (haveprec)
68         (void) asprintf(&b, f, precision, func);
69     else
70         (void) asprintf(&b, f, func);
71     if (b) {
72         (void) fputs(b, stdout);
73         free(b);
74     }
75     if (dollar)
76         *f = '$';
77 }

78 _NOTE(CONSTCOND) } while (0)

79 static int      asciicode(void);
80 static char     *dofORMAT(char *, int *);
81 static int      escape(char *, int, size_t *);
82 static int      getchr(void);
83 static int      getfloating(long double *, int);
84 static int      getint(int *);
85 static int      getnum(intmax_t *, uintmax_t *, int);
86 static int      *getstr(void);
87 static char     **mknum(char *, char);
88 static void     usage(void);

89 static const char digits[] = "0123456789";

90 static int      myargc;
91 static char     **myargv;
92 static char     **gargv;
93 static char     **maxargv;
```

new/usr/src/cmd/printf/printf.c

2

```
58     }
59     if (havewidth)
60         if (haveprec)
61             (void) asprintf(&b, f, fieldwidth, precision, func);
62         else
63             (void) asprintf(&b, f, fieldwidth, func);
64     else if (haveprec)
65         (void) asprintf(&b, f, precision, func);
66     else
67         (void) asprintf(&b, f, func);
68     if (b) {
69         (void) fputs(b, stdout);
70         free(b);
71     }
72     if (dollar)
73         *f = '$';
74 }

75 _NOTE(CONSTCOND) } while (0)

76 static int      asciicode(void);
77 static char     *dofORMAT(char *, int *);
78 static int      escape(char *, int, size_t *);
79 static int      getchr(void);
80 static int      getfloating(long double *, int);
81 static int      getint(int *);
82 static int      getnum(intmax_t *, uintmax_t *, int);
83 static int      *getstr(void);
84 static char     **mknum(char *, char);
85 static void     usage(void);

86 static const char digits[] = "0123456789";

87 static int      myargc;
88 static char     **myargv;
89 static char     **gargv;
90 static char     **maxargv;

91 int
92 main(int argc, char *argv[])
93 {
94     size_t len;
95     int end, rval;
96     int chopped, end, rval;
97     char *format, *fmt, *start;
98     (void) setlocale(LC_ALL, "");

99     argv++;
100    argc--;
101    argc--;

102    /*
103     * POSIX says: Standard utilities that do not accept options,
104     * but that do accept operands, shall recognize "--" as a
105     * first argument to be discarded.
106     */
107    if (argc && strcmp(argv[0], "--") == 0) {
108        argc--;
109        argv++;
110    }

111    if (argc < 1) {
112        usage();
113        return (1);
114    }

115    /*
116     */

117    /*
118     */
```

```

119     * Basic algorithm is to scan the format string for conversion
120     * specifications -- once one is found, find out if the field
121     * width or precision is a '*'; if it is, gather up value. Note,
122     * format strings are reused as necessary to use up the provided
123     * arguments, arguments of zero/null string are provided to use
124     * up the format string.
125     */
126
127     fmt = format = *argv;
128     (void) escape(fmt, 1, &len); /* backslash interpretation */
129     chopped = escape(fmt, 1, &len); /* backslash interpretation */
130     rval = end = 0;
131     argv = ++argv;
132
133     for (;;) {
134         maxargv = argv;
135         char **maxargv = argv;
136
137         myargv = argv;
138         for (myargc = 0; argv[myargc]; myargc++)
139             /* nop */;
140         start = fmt;
141         while (fmt < format + len) {
142             if (fmt[0] == '%') {
143                 (void) fwrite(start, 1, PTRDIFF(fmt, start),
144                               stdout);
145                 if (fmt[1] == '%') {
146                     /* %% prints a % */
147                     (void) putchar('%');
148                     fmt += 2;
149                 } else {
150                     fmt = doformat(fmt, &rval);
151                     if (fmt == NULL)
152                         return (1);
153                     end = 0;
154                 }
155                 start = fmt;
156             } else
157                 fmt++;
158             if (argv > maxargv)
159                 maxargv = argv;
160         }
161         argv = maxargv;
162
163         if (end == 1) {
164             warnx1(_("missing format character"), NULL, NULL);
165             return (1);
166         }
167         (void) fwrite(start, 1, PTRDIFF(fmt, start), stdout);
168         if (!*argv)
169             if (chopped || !*argv)
170                 return (rval);
171
172     }/* NOTREACHED */
173
174 static char *
175 doformat(char *fmt, int *rval)
176 doformat(char *start, int *rval)
177 {
178     static const char skip1[] = "#'-' 0";
179     static const char skip2[] = "0123456789";
180     char *fmt;

```

```

179     int fieldwidth, haveprec, havewidth, mod_ldbl, precision;
180     char convch, nextch;
181     char *start;
182     char **fargv;
183     char *dptr;
184     int l;
185
186     start = alloca(strlen(fmt) + 1);
187     fmt = start + 1;
188
189     dptr = start;
190     *dptr++ = '%';
191     *dptr = 0;
192
193     fmt++;
194
195     /* look for "n$" field index specifier */
196     l = strspn(fmt, digits);
197     if ((l > 0) && (fmt[l] == '$')) {
198         int idx = atoi(fmt);
199         fmt += strspn(fmt, skip2);
200         if ((*fmt == '$') && (fmt != (start + l))) {
201             int idx = atoi(start + l);
202             if (idx <= myargc) {
203                 argv = &myargv[idx - 1];
204             } else {
205                 argv = &myargv[myargc];
206             }
207             if (argv > maxargv) {
208                 maxargv = argv;
209             }
210             fmt += l + 1;
211
212             /* save format argument */
213             fargv = argv;
214             start = fmt;
215             fmt++;
216         } else {
217             fargv = NULL;
218             fmt = start + l;
219         }
220
221         /* skip to field width */
222         while (strchr(skip1, *fmt) != NULL) {
223             *dptr++ = *fmt++;
224             *dptr = 0;
225         }
226
227         fmt += strspn(fmt, skip1);
228         if (*fmt == '*') {
229             fmt++;
230             l = strspn(fmt, digits);
231             if ((l > 0) && (fmt[l] == '$')) {
232                 int idx = atoi(fmt);
233                 if (idx <= myargc) {
234                     argv = &myargv[idx - 1];
235                 } else {
236                     argv = &myargv[myargc];
237                 }
238                 fmt += l + 1;
239
240                 if (getint(&fieldwidth))
241                     return (NULL);
242             }
243         }
244     }

```

```

237         if (gargv > maxargv) {
238             maxargv = gargv;
239         }
240         havewidth = 1;
241
242         *dptr++ = '*';
243         *dptr = 0;
244     } else {
245         havewidth = 0;
246
247         /* skip to possible '.', get following precision */
248         while (isdigit(*fmt)) {
249             *dptr++ = *fmt++;
250             *dptr = 0;
251             fmt += strspn(fmt, skip2);
252         }
253
254         if (*fmt == '.') {
255             /* precision present? */
256             fmt++;
257             *dptr++ = '.';
258
259             ++fmt;
260             if (*fmt == '*') {
261
262                 fmt++;
263                 l = strspn(fmt, digits);
264                 if ((l > 0) && (fmt[l] == '$')) {
265                     int idx = atoi(fmt);
266                     if (idx <= myargc) {
267                         gargv = &myargv[idx - 1];
268                     } else {
269                         gargv = &myargv[myargc];
270                     }
271                     fmt += l + 1;
272
273                     if (getint(&precision))
274                         return (NULL);
275                     if (gargv > maxargv) {
276                         maxargv = gargv;
277                     }
278                     haveprec = 1;
279                     *dptr++ = '*';
280                     *dptr = 0;
281                     ++fmt;
282                 } else {
283                     haveprec = 0;
284
285                     /* skip to conversion char */
286                     while (isdigit(*fmt)) {
287                         *dptr++ = *fmt++;
288                         *dptr = 0;
289                         fmt += strspn(fmt, skip2);
290                     }
291                 }
292             haveprec = 0;
293             if (!*fmt) {
294                 warnx(_("missing format character"), NULL, NULL);
295             }
296             *dptr++ = *fmt;
297             *dptr = 0;

```

```

299
300     /* Look for a length modifier.  POSIX doesn't have these, so
301      * we only support them for floating-point conversions, which
302      * are extensions.  This is useful because the L modifier can
303      * be used to gain extra range and precision, while omitting
304      * it is more likely to produce consistent results on different
305      * architectures.  This is not so important for integers
306      * because overflow is the only bad thing that can happen to
307      * them, but consider the command printf %a 1.1
308
309     if (*fmt == 'L') {
310         mod_ldbl = 1;
311         fmt++;
312         if (!strchr("aAeEfFgG", *fmt)) {
313             warnx2(_("bad modifier L for %%c"), *fmt, NULL);
314             return (NULL);
315         }
316     } else {
317         mod_ldbl = 0;
318     }
319
320     /* save the current arg offset, and set to the format arg */
321     if (fargv != NULL) {
322         gargv = fargv;
323     }
324
325     convch = *fmt;
326     nextch = *++fmt;
327
328     *fmt = '\0';
329     switch (convch) {
330     case 'b':
331         size_t len;
332         char *p;
333         int getout;
334
335         p = strdup(getstr());
336         if (p == NULL) {
337             warnx2("%s", strerror(ENOMEM), NULL);
338             return (NULL);
339         }
340         getout = escape(p, 0, &len);
341         (void) fputs(p, stdout);
342         *(fmt - 1) = 's';
343         PF(start, p);
344         *(fmt - 1) = 'b';
345         free(p);
346
347         if (getout)
348             exit(*rval);
349         break;
350     case 'c':
351         char p;
352         p = getch();
353         PF(start, p);
354         break;
355     case 's':
356         const char *p;
357
358         p = getstr();
359         PF(start, p);

```

```

360         break;
361     }
362     case 'd': case 'i': case 'o': case 'u': case 'x': case 'X': {
363         char *f;
364         intmax_t val;
365         uintmax_t uval;
366         int signedconv;
367
368         signedconv = (convch == 'd' || convch == 'i');
369         if ((f = mknum(start, convch)) == NULL)
370             return (NULL);
371         if (getnum(&val, &uval, signedconv))
372             *rval = 1;
373         if (signedconv)
374             PF(f, val);
375         else
376             PF(f, uval);
377         break;
378     }
379     case 'e': case 'E':
380     case 'f': case 'F':
381     case 'g': case 'G':
382     case 'a': case 'A': {
383         long double p;
384
385         if (getfloating(&p, mod_ldbl))
386             *rval = 1;
387         if (mod_ldbl)
388             PF(start, p);
389         else
390             PF(start, (double)p);
391         break;
392     }
393     default:
394         warnx2(_("illegal format character %c"), convch, NULL);
395         return (NULL);
396     }
397     *fmt = nextch;
398
399     /* return the argv to the next element */
400     return (fmt);
401 }

```

unchanged_portion_omitted

```

429 static int
430 escape(char *fmt, int percent, size_t *len)
431 {
432     char *save, *store, c;
433     int value;
434
435     for (save = store = fmt; ((c = *fmt) != 0); ++fmt, ++store) {
436         if (c != '\\') {
437             *store = c;
438             continue;
439         }
440         switch (++fmt) {
441             case '\0':           /* EOS, user error */
442                 *store = '\\';
443                 *++store = '\0';
444                 *len = PTRDIFF(store, save);
445                 return (0);
446             case '\\':           /* backslash */
447             case '\'':           /* single quote */
448                 *store = *fmt;
449                 break;
450             case 'a':             /* bell/alert */

```

```

451             *store = '\a';
452             break;
453         case 'b':           /* backspace */
454             *store = '\b';
455             break;
456         case 'c':
457             if (!percent) {
458                 *store = '\0';
459                 *len = PTRDIFF(store, save);
460                 return (1);
461             }
462             *store = 'c';
463             break;
464         case 'f':           /* form-feed */
465             *store = '\f';
466             break;
467         case 'n':           /* newline */
468             *store = '\n';
469             break;
470         case 'r':           /* carriage-return */
471             *store = '\r';
472             break;
473         case 't':           /* horizontal tab */
474             *store = '\t';
475             break;
476         case 'v':           /* vertical tab */
477             *store = '\v';
478             break;
479             /* octal constant */
480         case '0': case '1': case '2': case '3':
481         case '4': case '5': case '6': case '7':
482             c = (!percent && *fmt == '0') ? 4 : 3;
483             for (value = 0;
484                  c-- && *fmt >= '0' && *fmt <= '7'; ++fmt) {
485                 value <= 3;
486                 value += *fmt - '0';
487             }
488             --fmt;
489             if (percent && value == '%') {
490                 *store++ = '%';
491                 *store = '%';
492             } else
493                 *store = (char)value;
494             break;
495         default:
496             *store = *fmt;
497             break;
498         }
499     }
500     *store = '\0';
501     *len = PTRDIFF(store, save);
502     return (0);
503 }

```

unchanged_portion_omitted

```
new/usr/src/pkg/manifests/system-test-utiltest.mf
```

```
1
```

```
*****
```

```
1236 Fri May 9 12:12:12 2014
```

```
new/usr/src/pkg/manifests/system-test-utiltest.mf  
4818 printf(1) should support n$ width and precision specifiers
```

```
*****
```

```
1 #  
2 # This file and its contents are supplied under the terms of the  
3 # Common Development and Distribution License (" CDDL"), version 1.0.  
4 # You may only use this file in accordance with the terms of version  
5 # 1.0 of the CDDL.  
6 #  
7 # A full copy of the text of the CDDL should have accompanied this  
8 # source. A copy of the CDDL is also available via the Internet at  
9 # http://www.illumos.org/license/CDDL.  
10 #  
  
12 #  
13 # Copyright (c) 2012 by Delphix. All rights reserved.  
14 # Copyright 2014, OmniTI Computer Consulting, Inc. All rights reserved.  
15 #  
  
17 set name=pkg.fmri value=(pkg:/system/test/utiltest@$(PKGVERS)  
18 set name=pkg.description value="Miscellaneous Utility Unit Tests"  
19 set name=pkg.summary value="Utility Unit Test Suite"  
20 set name=info.classification \  
21 value=org.opensolaris.category.2008:Development/System  
22 set name=variant.arch value=$(ARCH)  
23 dir path=opt/util-tests  
24 dir path=opt/util-tests/bin  
25 dir path=opt/util-tests/runfiles  
26 dir path=opt/util-tests/tests  
27 file path=opt/util-tests/README mode=0444  
28 file path=opt/util-tests/bin/utiltest mode=0555  
29 file path=opt/util-tests/runfiles/default.run mode=0444  
30 file path=opt/util-tests/tests/printf_test mode=0555  
31 license lic_CDDL license=lic_CDDL  
32 depend fmri=system/test/testrunner type=require
```

```
new/usr/src/test/Makefile
```

```
1
```

```
*****
```

```
6 13 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/Makefile
```

```
4854 printf(1) doesn't support %b and \c properly
```

```
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
```

```
10 #
```

```
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
```

```
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
```

```
15 #
```

```
17 .PARALLEL: $(SUBDIRS)
```

```
19 SUBDIRS = os-tests test-runner util-tests zfs-tests
```

```
18 SUBDIRS = os-tests test-runner zfs-tests
```

```
21 include Makefile.com
```

```
new/usr/src/test/util-tests/Makefile
```

```
1
```

```
*****
```

```
552 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/util-tests/Makefile
```

```
4818 printf(1) should support n$ width and precision specifiers
```

```
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 #
15 .
16 .PARALLEL: ${SUBDIRS}
17
18 SUBDIRS = cmd runfiles tests doc
19
20 include ${SRC}/test/Makefile.com
```

```
new/usr/src/test/util-tests/cmd/Makefile
```

```
1
```

```
*****
```

```
847 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/util-tests/cmd/Makefile
```

```
4818 printf(1) should support n$ width and precision specifiers
```

```
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #

12 #
13 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
14 # Copyright (c) 2012 by Delphix. All rights reserved.
15 #

17 include $(SRC)/Makefile.master
18 include $(SRC)/test/Makefile.com

20 ROOTOPTPKG = $(ROOT)/opt/util-tests
21 ROOTBIN = $(ROOTOPTPKG)/bin

23 PROGS = utiltest

25 CMDS = $(PROGS:%= $(ROOTBIN)/%)
26 $(CMDS) := FILEMODE = 0555

28 all lint clean clobber:

30 install: $(CMDS)

32 $(CMDS): $(ROOTBIN)

34 $(ROOTBIN):
35     $(INS.dir)

37 $(ROOTBIN)/%: %.ksh
38     $(INS.rename)
```

new/usr/src/test/util-tests/cmd/utiltest.ksh

1

```
*****
1449 Fri May 9 12:12:12 2014
new/usr/src/test/util-tests/cmd/utiltest.ksh
4818 printf(1) should support n$ width and precision specifiers
*****
1#!/usr/bin/ksh

3#
4# This file and its contents are supplied under the terms of the
5# Common Development and Distribution License (" CDDL"), version 1.0.
6# You may only use this file in accordance with the terms of version
7# 1.0 of the CDDL.
8#
9# A full copy of the text of the CDDL should have accompanied this
10# source. A copy of the CDDL is also available via the Internet at
11# http://www.illumos.org/license/CDDL.
12#

14#
15# Copyright (c) 2012 by Delphix. All rights reserved.
16# Copyright 2014, OmniTI Computer Consulting, Inc. All rights reserved.
17# Copyright 2014 Garrett D'Amore <garrett@damore.org>
18#

20 export MY_TESTS="/opt/util-tests"
21 runner="/opt/test-runner/bin/run"

23 function fail
24 {
25     echo $1
26     exit ${2:-1}
27 }

29 function find_runfile
30 {
31     typeset distro=
32     if [[ -d /opt/delphix && -h /etc/delphix/version ]]; then
33         distro=delphix
34     elif [[ 0 -ne $(grep -c OpenIndiana /etc/release 2>/dev/null) ]]; then
35         distro=openindiana
36     elif [[ 0 -ne $(grep -c OmniOS /etc/release 2>/dev/null) ]]; then
37         distro=omniios
38     elif [[ -f $MY_TESTS/runfiles/default.run ]]; then
39         # optional catch-all
40         distro=default
41     fi
43     [[ -n $distro ]] && echo $MY_TESTS/runfiles/$distro.run
44 }

46 while getopts c: c; do
47     case $c in
48     'c')
49         runfile=$OPTARG
50         [[ -f $runfile ]] || fail "Cannot read file: $runfile"
51         ;;
52     esac
53 done
54 shift $((OPTIND - 1))

56 [[ -z $runfile ]] && runfile=$(find_runfile)
57 [[ -z $runfile ]] && fail "Couldn't determine distro"

59 $runner -c $runfile
61 exit $?
```

new/usr/src/test/util-tests/cmd/utiltest.ksh

2

```
new/usr/src/test/util-tests/doc/Makefile
```

```
1
```

```
*****
```

```
800 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/util-tests/doc/Makefile
```

```
4818 printf(1) should support n$ width and precision specifiers
```

```
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #

12 #
13 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
14 # Copyright (c) 2012 by Delphix. All rights reserved.
15 #

17 include $(SRC)/Makefile.master

19 READMES = README

21 ROOTOPTPKG = $(ROOT)/opt/util-tests

23 FILES = $(READMES:%=$(ROOTOPTPKG)/%)
24 $(FILES) := FILEMODE = 0444

26 all: $(READMES)

28 install: $(ROOTOPTPKG) $(FILES)

30 clean lint clobber:

32 $(ROOTOPTPKG):
33     $(INS.dir)

35 $(ROOTOPTPKG)/%: %
36     $(INS.file)
```

```
new/usr/src/test/util-tests/doc/README
```

```
1
```

```
*****  
2180 Fri May 9 12:12:12 2014  
new/usr/src/test/util-tests/doc/README  
4818 printf(1) should support n$ width and precision specifiers  
*****
```

```
1 #  
2 # This file and its contents are supplied under the terms of the  
3 # Common Development and Distribution License (" CDDL"), version 1.0.  
4 # You may only use this file in accordance with the terms of version  
5 # 1.0 of the CDDL.  
6 #  
7 # A full copy of the text of the CDDL should have accompanied this  
8 # source. A copy of the CDDL is also available via the Internet at  
9 # http://www.illumos.org/license/CDDL.  
10 #  
12 #  
13 # Copyright 2014 Garrett D'Amore <garrett@damore.org>  
14 # Copyright (c) 2012 by Delphix. All rights reserved.  
15 #
```

17 Utils Unit Test Suite README

```
19 1. What the Utils Unit Test Suite tests  
20 2. Building and installing the Utils Unit Test Suite  
21 3. Running the Utils Unit Test Suite  
22 4. Test results
```

```
24 -----
```

26 1. What the Utils Unit Test Suite tests

```
28 The Utils unit test suite is for testing standard shell / POSIX utilities.  
29 For example utilities such as "printf" are tested.
```

31 2. Building and installing the Utils Unit Test Suite

```
33 The Utils Unit Test Suite runs under the testrunner framework (which can be  
34 installed as pkg:/system/test/testrunner). To build both the Utils Unit Test  
35 Suite and the testrunner without running a full nightly:
```

```
37     build_machine$ bldenv [-d] <your_env_file>  
38     build_machine$ cd $SRC/test  
39     build_machine$ dmake install  
40     build_machine$ cd $SRC/pkg  
41     build_machine$ dmake install
```

```
43 Then set the publisher on the test machine to point to your repository and  
44 install the Utils Unit Test Suite.
```

```
46     test_machine# pkg install pkg:/system/test/utiltest
```

```
48 Note, the framework will be installed automatically, as the Utils Unit Test  
49 Suite depends on it.
```

51 3. Running the Utils Unit Test Suite

```
53 The pre-requisites for running the OS Unit Test Suite are:  
54     - Any user may perform these tests.
```

```
56 Once the pre-requisites are satisfied, simply run the ostest script:
```

```
58     test_machine$ /opt/util-tests/bin/utiltest
```

60 4. Test results

```
new/usr/src/test/util-tests/doc/README
```

```
2
```

```
62 While the OS Unit Test Suite is running, one informational line is printed at  
63 the end of each test, and a results summary is printed at the end of the run.  
64 The results summary includes the location of the complete logs, which is of the  
65 form /var/tmp/test_results/<ISO 8601 date>.
```

```
new/usr/src/test/util-tests/runfiles/Makefile
```

```
1
```

```
*****  
909 Fri May 9 12:12:12 2014  
new/usr/src/test/util-tests/runfiles/Makefile  
4818 printf(1) should support n$ width and precision specifiers  
*****
```

```
1 #  
2 # This file and its contents are supplied under the terms of the  
3 # Common Development and Distribution License (" CDDL"), version 1.0.  
4 # You may only use this file in accordance with the terms of version  
5 # 1.0 of the CDDL.  
6 #  
7 # A full copy of the text of the CDDL should have accompanied this  
8 # source. A copy of the CDDL is also available via the Internet at  
9 # http://www.illumos.org/license/CDDL.  
10 #  
12 #  
13 # Copyright (c) 2012 by Delphix. All rights reserved.  
14 # Copyright 2014, OmniTI Computer Consulting, Inc. All rights reserved.  
15 # Copyright 2014 Garrett D'Amore <garrett@damore.org>  
16 #  
18 include $(SRC)/Makefile.master  
20 SRCS = default.run  
22 ROOTOPTPKG = $(ROOT)/opt/util-tests  
23 RUNFILES = $(ROOTOPTPKG)/runfiles  
25 CMDS = $(SRCS:%= $(RUNFILES) /%)  
26 $(CMDS) := FILEMODE = 0444  
28 all: $(SRCS)  
30 install: $(CMDS)  
32 clean lint clobber:  
34 $(CMDS): $(RUNFILES) $(SRCS)  
36 $(RUNFILES):  
37     $(INS.dir)  
39 $(RUNFILES) /%: %  
40     $(INS.file)
```

```
new/usr/src/test/util-tests/runfiles/default.run
```

```
1
```

```
*****
```

```
654 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/util-tests/runfiles/default.run  
4818 printf(1) should support n$ width and precision specifiers
```

```
*****
```

```
1 #  
2 # This file and its contents are supplied under the terms of the  
3 # Common Development and Distribution License (" CDDL"), version 1.0.  
4 # You may only use this file in accordance with the terms of version  
5 # 1.0 of the CDDL.  
6 #  
7 # A full copy of the text of the CDDL should have accompanied this  
8 # source. A copy of the CDDL is also available via the Internet at  
9 # http://www.illumos.org/license/CDDL.  
10 #  
  
12 #  
13 # Copyright (c) 2012 by Delphix. All rights reserved.  
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>  
15 #  
  
17 [DEFAULT]  
18 pre =  
19 verbose = False  
20 quiet = False  
21 timeout = 60  
22 post =  
23 outputdir = /var/tmp/test_results  
  
25 [/opt/util-tests/tests/printf_test]
```

```
new/usr/src/test/util-tests/tests/Makefile
```

```
1
```

```
*****
```

```
567 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/util-tests/tests/Makefile
```

```
4818 printf(1) should support n$ width and precision specifiers
```

```
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #

12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 #

17 SUBDIRS = printf
19 include $(SRC)/test/Makefile.com
```

```
new/usr/src/test/util-tests/tests/printf/Makefile
```

```
1
```

```
*****
```

```
867 Fri May 9 12:12:12 2014
```

```
new/usr/src/test/util-tests/tests/printf/Makefile
4818 printf(1) should support n$ width and precision specifiers
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #

12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 #

17 include $(SRC)/cmd/Makefile.cmd
18 include $(SRC)/test/Makefile.com

20 PROG = printf_test

22 ROOTOPTPKG = $(ROOT)/opt/util-tests
23 TESTDIR = $(ROTOPTPKG)/tests

25 CMDS = $(PROG:%= $(TESTDIR)/%)
26 $(CMDS) := FILEMODE = 0555

28 all lint clean clobber:

30 install: all $(CMDS)

32 $(CMDS): $(TESTDIR) $(PROG).ksh

34 $(TESTDIR):
35     $(INS.dir)

37 $(TESTDIR)/%: %.ksh
38     $(INS.rename)
```

```

new/usr/src/test/util-tests/tests/printf/printf_test.ksh
*****
4477 Fri May 9 12:12:12 2014
new/usr/src/test/util-tests/tests/printf/printf_test.ksh
4854 printf(1) doesn't support %b and \c properly
4818 printf(1) should support n$ width and precision specifiers
*****
1 #! /usr/bin/ksh
2 #
3 #
4 # This file and its contents are supplied under the terms of the
5 # Common Development and Distribution License (" CDDL"), version 1.0.
6 # You may only use this file in accordance with the terms of version
7 # 1.0 of the CDDL.
8 #
9 # A full copy of the text of the CDDL should have accompanied this
10 # source. A copy of the CDDL is also available via the Internet at
11 # http://www.illumos.org/license/CDDL.
12 #

14 #
15 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
16 #

18 PRINTF=${PRINTF:=`/usr/bin/printf`}
20 test_start() {
21     print "TEST STARTING ${1}: ${2}"
22 }

24 test_pass() {
25     print "TEST PASS: ${1}"
26 }

28 test_fail() {
29     print "TEST FAIL: ${1}: ${2}"
30     exit -1
31 }

33 checkrv() {
34     if [[ $? -ne 0 ]]; then
35         test_fail $1 "exit failure"
36     fi
37 }

39 compare() {
40     if [[ "$2" != "$3" ]]; then
41         test_fail $1 "compare mismatch, got [$2] expected [$3]"
42     fi
43 }

45 typeset -A tests=()

48 typeset -A tests[01]=()
49 tests[01][desc]="hexadecimal lowercase"
50 tests[01][format]='%04x'
51 tests[01][args]="255"
52 tests[01][result]="00ff"

54 typeset -A tests[02]=()
55 tests[02][desc]="hexadecimal 32-bit"
56 tests[02][format]='%08x'
57 tests[02][args]="'65537'"
58 tests[02][result]=00010001

60 typeset -A tests[03]=()

```

1

```

new/usr/src/test/util-tests/tests/printf/printf_test.ksh
*****
61 tests[03][desc]="multiple arguments"
62 tests[03][format]='%d %s '
63 tests[03][args]="1 one 2 two 3 three"
64 tests[03][result]='1 one 2 two 3 three '

66 typeset -A tests[04]=()
67 tests[04][desc]="variable position parameters"
68 tests[04][format]='%2$s %1$d '
69 tests[04][args]="1 one 2 two 3 three"
70 tests[04][result]='one 1 two 2 three 3 '

72 typeset -A tests[05]=()
73 tests[05][desc]="width"
74 tests[05][format]='%10s'
75 tests[05][args]="abcdef"
76 tests[05][result]='      abcdef'

78 typeset -A tests[06]=()
79 tests[06][desc]="width and precision"
80 tests[06][format]='%10.3s'
81 tests[06][args]="abcdef"
82 tests[06][result]='      abc'

84 typeset -A tests[07]=()
85 tests[07][desc]="variable width and precision"
86 tests[07][format]='%.*.*s'
87 tests[07][args]="10 3 abcdef"
88 tests[07][result]='      abc'

90 typeset -A tests[08]=()
91 tests[08][desc]="variable position width and precision"
92 tests[08][format]='%2$*1$.*3$s'
93 tests[08][args]="10 abcdef 3"
94 tests[08][result]='      abc'

96 typeset -A tests[09]=()
97 tests[09][desc]="multi variable position width and precision"
98 tests[09][format]='%2$*1$.*3$'
99 tests[09][args]="10 abcdef 3 5 xyz 1"
100 tests[09][result]='      abc      x'

102 typeset -A tests[10]=()
103 tests[10][desc]="decimal from hex"
104 tests[10][format]='%d'
105 tests[10][args]="0x1000 0XA"
106 tests[10][result]='4096 10 '

108 typeset -A tests[11]=()
109 tests[11][desc]="negative dec (64-bit)"
110 tests[11][format]='%x'
111 tests[11][args]="-1"
112 tests[11][result]='ffffffffffffffffff'

114 typeset -A tests[12]=()
115 tests[12][desc]="float (basic)"
116 tests[12][format]='%f'
117 tests[12][args]="3.14"
118 tests[12][result]='3.140000'

120 typeset -A tests[12]=()
121 tests[12][desc]="float precision"
122 tests[12][format]='%.2f'
123 tests[12][args]="3.14159"
124 tests[12][result]='3.14'

126 typeset -A tests[13]=()

```

2

```

new/usr/src/test/util-tests/tests/printf/printf_test.ksh

127 tests[13][desc]="left justify"
128 tests[13][format]=`%-5d'
129 tests[13][args]="'45"
130 tests[13][result]='45      '

132 typeset -A tests[14]()
133 tests[14][desc]="newlines"
134 tests[14][format]=`%s\n%s\n%s'
135 tests[14][args]="one two three"
136 tests[14][result]='one
137 two
138 three'

140 typeset -A tests[15]()
141 tests[15][desc]="embedded octal escape"
142 tests[15][format]='%s\41%s'
143 tests[15][args]="one two"
144 tests[15][result]='one!two'

146 typeset -A tests[16]()
147 tests[16][desc]="backslash string (%b)"
148 tests[16][format]='%b'
149 tests[16][args]]='\0101\0102\0103'
150 tests[16][result]='ABC'

152 typeset -A tests[17]()
153 tests[17][desc]="backslash c in %b"
154 tests[17][format]='%b%'
155 tests[17][args]]='\0101\cone two'
156 tests[17][result]='A'

158 typeset -A tests[18]()
159 tests[18][desc]="backslash octal in format"
160 tests[18][format]='HI\1120K\0112tabbed\11again'
161 tests[18][args]=
162 tests[18][result]='HIJOK          2tabbed again'

164 typeset -A tests[19]()
165 tests[19][desc]="backslash octal in %b"
166 tests[19][format]="%b"
167 tests[19][args]='HI\0112K\011tabbed'
168 tests[19][result]='HIJK tabbed'

170 typeset -A tests[20]()
171 tests[20][desc]="numeric %d and ASCII conversions"
172 tests[20][format]='%d '
173 tests[20][args]="3 +3 -3 \"3 \"+ \"-"
174 tests[20][result]='3 3 -3 51 43 45 '

176 #debug=yes

178 for i in "${!tests[@]}"; do
179     t=test_$i
180     desc=${tests[$i][desc]}
181     format=${tests[$i][format]}
182     args=${tests[$i][args]}
183     result=${tests[$i][result]}
184
185     test_start $t "${tests[$i][desc]}"
186     [[ -n "$debug" ]] && echo $PRINTF "$format" "${args[@]}"
187     comp=$(PRINTF "$format" ${args[@]})
188     checkrv $t
189     [[ -n "$debug" ]] && echo "got [$comp]"
190     good=$result
191     compare $t "$comp" "$good"
192     test_pass $t

```

3

```

new/usr/src/test/util-tests/tests/printf/printf_test.ksh

```

```

193 done

```

4