

```

*****
3229 Tue Dec 6 07:59:24 2016
new/usr/src/cmd/oamuser/user/Makefile
7648 useradd/mod commands operate on /home special filesystem
*****
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 #
21 #
22 # Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013 RackTop Systems.
24 # Copyright (c) 2013 Gary Mills
25 #
26 # cmd/oamuser/user/Makefile
27 #

29 DEFAULTFILES= useradd.dfl

31 include ../../Makefile.cmd

33 GREP=          grep

35 USERADD=       useradd
36 USERDEL=       userdel
37 USERMOD=       usermod
38 ROLEADD=       roleadd
39 ROLEDEL=       roledel
40 ROLEMOD=       rolemod

42 SBINPROG=      $(USERADD) $(USERDEL) $(USERMOD)
43 #
44 # Removing sysadm: deleted $(SYSADMPROG) from this target.
45 #
46 PROG=          $(SBINPROG)
47 PRODUCT=       $(PROG)

49 ADD_OBJ=       useradd.o    homedir.o    groups.o    call_pass.o \
50                userdefs.o  messages.o  val_lgrp.o  funcs.o \
51                val_lprj.o  proj.o

53 DEL_OBJ=       userdel.o    call_pass.o  homedir.o  isbusy.o \
54                groups.o    messages.o  funcs.o    proj.o

56 MOD_OBJ=       usermod.o    movedir.o   groups.o   homedir.o \
57                call_pass.o isbusy.o   userdefs.o \
58                messages.o val_lgrp.o  funcs.o    val_lprj.o \
59                proj.o

61 OBJECTS=       $(ADD_OBJ)    $(DEL_OBJ)  $(MOD_OBJ)

```

```

63 SRCS=          $(OBJECTS:.o=.c)

65 LIBDIR=        ../lib
66 LIBUSRGRP=     $(LIBDIR)/lib.a
67 LIBADM=        -ladm
68 LOCAL=         ../inc
69 HERE=          .
70 LINTFLAGS=     -u

72 ROOTSKEL=      $(ROOTETC)/skel
73 INSSBINPROG=   $(SBINPROG:%=$(ROOTUSRSBIN)/%)
74 INSSKELFILE=   $(SKELFILE:%=$(ROOTSKEL)/%)

76 CPPFLAGS=      -I$(HERE) -I$(LOCAL) $(CPPFLAGS.master)
77 CPPFLAGS +=    -DDEFAULT_USERADD=\"etc/default/useradd\"
78 CERRWARN +=    -gcc=-Wno-implicit-function-declaration

80 $(INSSBINPROG) := FILEMODE = 0555
81 $(INSSYSADMPROG) := FILEMODE = 0500
82 $(INSSKELFILE) := FILEMODE = 0644

84 $(USERADD) :=  OBJS = $(ADD_OBJ)
85 $(USERADD) :=  LIBS = $(LIBUSRGRP)
86 $(USERADD) :=  LDLIBS += -lcmdutils

88 $(USERDEL) :=  OBJS = $(DEL_OBJ)
89 $(USERDEL) :=  LIBS = $(LIBUSRGRP)

91 $(USERMOD) :=  OBJS = $(MOD_OBJ)
92 $(USERMOD) :=  LIBS = $(LIBUSRGRP)

94 LDLIBS +=      -lbsd -lnsl -lsecdb -lproject -lzfs -ltsol

96 .PARALLEL:    $(OBJECTS)

98 all:          $(PRODUCT)

100 $(PROG):      $$$(OBJ) $$$(LIBS)
101               $(LINK.c) $(OBJ) -o $@ $(LIBS) $(LDLIBS)
102               $(POST_PROCESS)

104 $(USERADD):   $(ADD_OBJ)
105 $(USERMOD):   $(MOD_OBJ)
106 $(USERDEL):   $(DEL_OBJ)

108 install:     all $(ROOTETCDEFAULTFILES) .WAIT \
109               $(ROOTSKEL) $(INSSBINPROG) $(INSSKELFILE)
110               $(RM) $(ROOTUSRSBIN)/$(ROLEADD)
111               $(LN) $(ROOTUSRSBIN)/$(USERADD) $(ROOTUSRSBIN)/$(ROLEADD)
112               $(RM) $(ROOTUSRSBIN)/$(ROLEDEL)
113               $(LN) $(ROOTUSRSBIN)/$(USERDEL) $(ROOTUSRSBIN)/$(ROLEDEL)
114               $(RM) $(ROOTUSRSBIN)/$(ROLEMOD)
115               $(LN) $(ROOTUSRSBIN)/$(USERMOD) $(ROOTUSRSBIN)/$(ROLEMOD)

117 clean:
118               $(RM) $(OBJECTS)

120 lint:        lint_SRCS

122 include ../../Makefile.targ

```

new/usr/src/cmd/oamuser/user/funcs.h

1

```
*****
1698 Tue Dec 6 07:59:24 2016
new/usr/src/cmd/oamuser/user/funcs.h
7648 useradd/mod commands operate on /home special filesystem
*****
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, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23 * Copyright 1999,2002-2003 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 */

27 #ifndef _FUNCS_H
28 #define _FUNCS_H

30 #ifdef __cplusplus
31 extern "C" {
32 #endif

34 #define CMD_PREFIX_USER "user"

36 #define AUTH_SEP      ", "
37 #define PROF_SEP     ", "
38 #define ROLE_SEP     ", "

40 #ifndef DEFAULT_USERADD
41 #define DEFAULT_USERADD "/etc/default/useradd"
42 #endif

44 #define MAX_TYPE_LENGTH 64

46 char *getusertype(char *cmdname);

48 int is_role(char *usertype);

50 void change_key(const char *, char *);
51 void addkey_args(char **, int *);
52 char *getsetdefval(const char *, char *);

54 extern int nkeys;

56 /* create_home() or rm_files() flags */
57 #define MANAGE_ZFS_OPT "MANAGE_ZFS="
58 #define MANAGE_ZFS    1

60 /* home parent settings for create_home() */
61 #define HOME_PARENT_OPT "HOME_PARENT="
```

new/usr/src/cmd/oamuser/user/funcs.h

2

```
62 #define HOME_PARENT      "/home"

64 #ifdef __cplusplus
65 }
_____unchanged_portion_omitted_____
```



```

120         if (zfs_mount(zhp, NULL, 0) != 0) {
121             errmsg(M_OOPS, "mount the home directory",
122                 libzfs_error_description(g_zfs));
123             (void) zfs_destroy(zhp, B_FALSE);
124             zfs_close(zhp);
125             libzfs_fini(g_zfs);
126             g_zfs = NULL;
127             return (EX_HOMEDIR);
128         }
129
130     zfs_close(zhp);
131
132     if (chmod(homedir,
133         S_IRWXU|S_IRGRP|S_IXGRP|S_IROTH|S_IXOTH) != 0) {
134         errmsg(M_OOPS,
135             "change permissions of home directory",
136             strerror(errno));
137         libzfs_fini(g_zfs);
138         g_zfs = NULL;
139         return (EX_HOMEDIR);
140     }
141
142     created_fs = 1;
143 } else {
144     errmsg(M_NO_ZFS_MOUNTPOINT, pdir);
145 }
146
147
148 if (!created_fs) {
149     if (mkdir(homedir, S_IRWXU|S_IRGRP|S_IXGRP|S_IROTH|S_IXOTH)
150         != 0) {
151         errmsg(M_OOPS, "create the home directory",
152             strerror(errno));
153         if (g_zfs != NULL) {
154             libzfs_fini(g_zfs);
155             g_zfs = NULL;
156         }
157         return (EX_HOMEDIR);
158     }
159 }
160
161 if (chown(homedir, uid, gid) != 0) {
162     errmsg(M_OOPS, "change ownership of home directory",
163         strerror(errno));
164     if (g_zfs != NULL) {
165         libzfs_fini(g_zfs);
166         g_zfs = NULL;
167     }
168     return (EX_HOMEDIR);
169 }
170
171 if (skeldir != NULL) {
172     /* copy the skel_dir into the home directory */
173     (void) sprintf(cmdbuf, "cd %s && find . -print | cpio -pd %s",
174         skeldir, homedir);
175
176     if (system(cmdbuf) != 0) {
177         errmsg(M_OOPS, "copy skeleton directory into home "
178             "directory", strerror(errno));
179         (void) rm_homedir(homedir, flags);
180         if (g_zfs != NULL) {
181             libzfs_fini(g_zfs);
182             g_zfs = NULL;
183         }
184         return (EX_HOMEDIR);
185     }

```

```

187     /* make sure contents in the home dirctory have correct owner */
188     (void) sprintf(cmdbuf,
189         "cd %s && find . -exec chown %ld:%ld {} \\;",
190         homedir, uid, gid);
191     if (system(cmdbuf) != 0) {
192         errmsg(M_OOPS,
193             "change owner and group of files home directory",
194             strerror(errno));
195         (void) rm_homedir(homedir, flags);
196         if (g_zfs != NULL) {
197             libzfs_fini(g_zfs);
198             g_zfs = NULL;
199         }
200         return (EX_HOMEDIR);
201     }
202
203 }
204 if (g_zfs != NULL) {
205     libzfs_fini(g_zfs);
206     g_zfs = NULL;
207 }
208 return (EX_SUCCESS);
209 }

```

unchanged portion omitted

new/usr/src/cmd/oamuser/user/useradd.dfl

1

1571 Tue Dec 6 07:59:24 2016

new/usr/src/cmd/oamuser/user/useradd.dfl

7648 useradd/mod commands operate on /home special filesystem

```
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 #
21 #
22 # Copyright (c) 2013 Gary Mills

24 # The EXCEED_TRAD indicates the action when the traditional login name
25 # length limit of eight characters is exceeded. The value "warning"
26 # means to issue a warning message and continue. This is the default.
27 # The value "error" means to issue an error message and terminate.
28 # The value "silent" means to continue without issuing any message.
29 #
30 EXCEED_TRAD=warning
31 #EXCEED_TRAD=error
32 #EXCEED_TRAD=silent

34 # The MANAGE_ZFS indicates if ZFS create/destroy operations
35 # should be performed by default when user passes -m flag to
36 # userdel/usermod or -r flag to userdel
37 MANAGE_ZFS=NO
38 #MANAGE_ZFS=YES

40 # The HOME_PARENT specifies base directory for new login home
41 # directories if /usr/sadm/defadduser file doesn't exist
42 HOME_PARENT=/home
```

```

*****
10481 Tue Dec 6 07:59:24 2016
new/usr/src/cmd/oamuser/user/userdefs.c
7648 useradd/mod commands operate on /home special filesystem
*****
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, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */

23 /*
24 * Copyright 2005 Sun Microsystems, Inc. All rights reserved.
25 * Use is subject to license terms.
26 */

28 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
29 /*      All Rights Reserved */

31 /*
32 * Copyright (c) 2013 RackTop Systems.
33 */

35 /*LINTLIBRARY*/

37 #include <sys/types.h>
38 #include <deflt.h>
39 #include <stdio.h>
40 #include <string.h>
41 #include <userdefs.h>
42 #include <user_attr.h>
43 #include <limits.h>
44 #include <stdlib.h>
45 #include <stddef.h>
46 #include <time.h>
47 #include <unistd.h>
48 #include "userdisp.h"
49 #include "funcs.h"
50 #include "messages.h"

52 /* Print out a NL when the line gets too long */
53 #define PRINTNL() \
54     if (outcount > 40) { \
55         outcount = 0; \
56         (void) fprintf(fp, "\n"); \
57     }

59 #define SKIPWS(ptr) while (*ptr && (*ptr == ' ' || *ptr == '\t')) ptr++

61 static char *dup_to_nl(char *);

```

```

63 static struct userdefs defaults = {
64     DEFPRID, DEFGROUP, DEFGNAME, DEFPARENT, DEFASKL,
65     DEFSQL, DEFINACT, DEFEXPIRE, DEFAUTH, DEFPROF,
66     DEFROLE, DEFPROJ, DEFPROJNAME, DEFLIMPRIV,
67     DEFDFLTPRIV, DEFLOCK_AFTER_RETRIES
68 };
unchanged_portion_omitted
110 #define NDEF (sizeof (tab) / sizeof (parent_t))
112 FILE *defptr; /* default file - fptr */

114 static char *
115 get_default_homeparent()
116 {
117     static char home_parent[PATH_MAX+1];
118
119     bzero(home_parent, PATH_MAX+1);
120     if (defopen(DEFAULT_USERADD) == 0) {
121         char *defptr;
122
123         if ((defptr = defread(HOME_PARENT_OPT)) != NULL) {
124             strncpy(home_parent, defptr, PATH_MAX);
125         }
126         (void) defopen((char *)NULL);
127     }
128
129     if (home_parent[0] == '\0') {
130         strcpy(home_parent, HOME_PARENT);
131     }
132     return (home_parent);
133 }

135 static const parent_t *
136 scan(char **start_p)
137 {
138     static int ind = NDEF - 1;
139     char *cur_p = *start_p;
140     int lastind = ind;
141
142     if (!*cur_p || *cur_p == '\n' || *cur_p == '#')
143         return (NULL);
144
145     /*
146      * The magic in this loop is remembering the last index when
147      * reentering the function; the entries above are also used to
148      * order the output to the default file.
149      */
150     do {
151         ind++;
152         ind %= NDEF;
153
154         if (strcmp(cur_p, tab[ind].name, tab[ind].nmsz) == 0) {
155             *start_p = cur_p + tab[ind].nmsz;
156             return (&tab[ind]);
157         }
158     } while (ind != lastind);
159
160     return (NULL);
161 }

163 /*
164 * getusrdef - access the user defaults file. If it doesn't exist,
165 * then returns default values of (values in userdefs.h):
166 * defrid = 100

```

```

167 *          defgroup = 1
168 *          defgname = other
169 *          defparent = /home
170 *          defskel = /usr/sadm/skel
171 *          defshell = /bin/sh
172 *          definact = 0
173 *          defexpire = 0
174 *          defauth = 0
175 *          defprof = 0
176 *          defrole = 0
177 *
178 *      If getusrdef() is unable to access the defaults file, it
179 *      returns a NULL pointer.
180 *
181 *      If user defaults file exists, then getusrdef uses values
182 *      in it to override the above values.
183 */

185 struct userdefs *
186 getusrdef(char *usertype)
187 {
188     char instr[512], *ptr;
189     const parsent_t *pe;

191     if (is_role(usertype)) {
192         if ((defptr = fopen(DEFROLEFILE, "r")) == NULL) {
193             defaults.defshell = DEFROLESHELL;
194             defaults.defprof = DEFROLEPROF;
195             defaults.defparent = get_default_homeparent();
196             return (&defaults);
197         }
198     } else {
199         if ((defptr = fopen(DEFFILE, "r")) == NULL) {
200             defaults.defparent = get_default_homeparent();
201             if ((defptr = fopen(DEFFILE, "r")) == NULL)
202                 return (&defaults);
203         }
204     }

205     while (fgets(instr, sizeof (instr), defptr) != NULL) {
206         ptr = instr;

208         SKIPWS(ptr);

210         if (*ptr == '#')
211             continue;

213         pe = scan(&ptr);

215         if (pe != NULL) {
216             switch (pe->type) {
217                 case INT:
218                     FIELD(&defaults, pe, int) =
219                         (int)strtol(ptr, NULL, 10);
220                     break;
221                 case PROJID:
222                     FIELD(&defaults, pe, projid_t) =
223                         (projid_t)strtol(ptr, NULL, 10);
224                     break;
225                 case STR:
226                     FIELD(&defaults, pe, char *) = dup_to_nl(ptr);
227                     break;
228             }
229         }
230     }

```

```

232         (void) fclose(defptr);

234         return (&defaults);
235     }
    _____
    unchanged_portion_omitted

```

13704 Tue Dec 6 07:59:25 2016

new/usr/src/man/man1m/useradd.1m

7648 useradd/mod commands operate on /home special filesystem

```

1  \" te
2  .\" Copyright (c) 2013 Gary Mills
3  .\" Copyright (c) 2008 Sun Microsystems, Inc. All Rights Reserved.
4  .\" Copyright 1989 AT&T
5  .\" The contents of this file are subject to the terms of the Common Development
6  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
7  .\" When distributing Covered Code, include this CDDL HEADER in each file and in
8  .TH USERADD 1M \"Dec 5, 2016\"
8  .TH USERADD 1M \"Apr 16, 2013\"
9  .SH NAME
10 useradd \- administer a new user login on the system
11 .SH SYNOPSIS
12 .LP
13 .nf
14 \fBuseradd\fR [\fB-A\fR \fIauthorization\fR [, \fIauthorization...\fR]]
15 [\fB-b\fR \fIbase_dir\fR] [\fB-c\fR \fIcomment\fR] [\fB-d\fR \fIidir\fR] [\fB-f\fR
16 [\fB-f\fR \fIinactive\fR] [\fB-g\fR \fIgroup\fR] [\fB-G\fR \fIgroup\fR [, \fB-k\fR
17 [\fB-K\fR \fIkey-value\fR] [\fB-m\fR \fIz\fR] [\fB-k\fR \fIIskel_dir\fR]
18 [\fB-P\fR \fIprofile\fR [, \fIprofile...\fR]] [\fB-R\fR \fIrole\fR [, \fIrole
19 [\fB-s\fR \fIshell\fR] [\fB-u\fR \fIuid\fR] [\fB-o\fR]] \fIlogin\fR
20 .fi
22 .LP
23 .nf
24 \fBuseradd\fR \fB-D\fR [\fB-A\fR \fIauthorization\fR [, \fIauthorization...\fR]]
25 [\fB-b\fR \fIbase_dir\fR] [\fB-s\fR \fIshell\fR] [\fB-k\fR \fIIskel_dir\fR]]
26 [\fB-f\fR \fIinactive\fR] [\fB-g\fR \fIgroup\fR] [\fB-K\fR \fIkey-value\fR]
27 [\fB-P\fR \fIprofile\fR [, \fIprofile...\fR]] [\fB-R\fR \fIrole\fR [, \fIrole
28 .fi
30 .SH DESCRIPTION
31 .LP
32 \fBuseradd\fR adds a new user to the \fB/etc/passwd\fR and \fB/etc/shadow\fR
33 and \fB/etc/user_attr\fR files. The \fB-A\fR and \fB-P\fR options respectively
34 assign authorizations and profiles to the user. The \fB-R\fR option assigns
35 roles to a user. The \fB-p\fR option associates a project with a user. The
36 \fB-K\fR option adds a \fIkey=value\fR pair to \fB/etc/user_attr\fR for the
37 user. Multiple \fIkey=value\fR pairs may be added with multiple \fB-K\fR
38 options.
39 .sp
40 .LP
41 \fBuseradd\fR also creates supplementary group memberships for the user
42 (\fB-G\fR option) and creates the home directory (\fB-m\fR option) for the user
43 if requested. The new login remains locked until the \fBpasswd\fR(1) command is
44 executed.
45 .sp
46 .LP
47 Specifying \fBuseradd\fR \fB-D\fR with the \fB-s\fR, \fB-k\fR, \fB-g\fR,
48 \fB-b\fR, \fB-f\fR, \fB-e\fR, \fB-A\fR, \fB-P\fR, \fB-p\fR, \fB-R\fR, or
49 \fB-K\fR option (or any combination of these options) sets the default values
50 for the respective fields. See the \fB-D\fR option, below. Subsequent
51 \fBuseradd\fR commands without the \fB-D\fR option use these arguments.
52 .sp
53 .LP
54 The system file entries created with this command have a limit of 2048
55 characters per line. Specifying long arguments to several options can exceed
56 this limit.
57 .sp
58 .LP
59 \fBuseradd\fR requires that usernames be in the format described in
60 \fBpasswd\fR(4). A warning message is displayed if these restrictions are not

```

```

61 met. See \fBpasswd\fR(4) for the requirements for usernames.
62 .LP
63 To change the action of \fBuseradd\fR when the traditional login name
64 length limit of eight characters is exceeded, edit the file
65 \fB/etc/default/useradd\fR by removing the \fB#\fR (pound sign) before the
66 appropriate \fBEXCEED_TRAD=\fR entry, and adding it before the others.
67 .SH OPTIONS
68 .LP
69 The following options are supported:
70 .sp
71 .ne 2
72 .na
73 \fB\fB-A\fR \fIauthorization\fR\fR
74 .ad
75 .sp .6
76 .RS 4n
77 One or more comma separated authorizations defined in \fBauth_attr\fR(4). Only
78 a user or role who has \fBgrant\fR rights to the authorization can assign it to
79 an account.
80 .RE
82 .sp
83 .ne 2
84 .na
85 \fB\fB-b\fR \fIbase_dir\fR
86 .ad
87 .sp .6
88 .RS 4n
89 The base directory for new login home directories (see the \fB-d\fR option
90 below. When a new user account is being created, \fIbase_dir\fR must already
91 exist unless the \fB-m\fR option or the \fB-d\fR option is also specified.
92 .RE
94 .sp
95 .ne 2
96 .na
97 \fB\fB-c\fR \fIcomment\fR\fR
98 .ad
99 .sp .6
100 .RS 4n
101 Any text string. It is generally a short description of the login, and is
102 currently used as the field for the user's full name. This information is
103 stored in the user's \fB/etc/passwd\fR entry.
104 .RE
106 .sp
107 .ne 2
108 .na
109 \fB\fB-d\fR \fIidir\fR\fR
110 .ad
111 .sp .6
112 .RS 4n
113 The home directory of the new user. It defaults to
114 \fIbase_dir\fR/\fIaccount_name\fR, where \fIbase_dir\fR is the base directory
115 for new login home directories and \fIaccount_name\fR is the new login name.
116 .RE
118 .sp
119 .ne 2
120 .na
121 \fB\fB-D\fR\fR
122 .ad
123 .sp .6
124 .RS 4n
125 Display the default values for \fBgroup\fR, \fBbase_dir\fR, \fBskel_dir\fR,
126 \fBshell\fR, \fBinactive\fR, \fBexpire\fR, \fBproj\fR, \fBprojname\fR and

```



```

127 \fBkey=value\fR pairs. When used with the \fB-g\fR, \fB-b\fR, \fB-f\fR,
128 \fB-e\fR, \fB-A\fR, \fB-P\fR, \fB-p\fR, \fB-R\fR, or \fB-K\fR options, the
129 \fB-D\fR option sets the default values for the specified fields. The default
130 values are:

```

```

131 .sp
132 .ne 2
133 .na
134 \fBgroup\fR
135 .ad
136 .sp .6
137 .RS 4n
138 \fBBother\fR (\fBGID\fR of 1)
139 .RE

```

```

141 .sp
142 .ne 2
143 .na
144 \fBbase_dir\fR
145 .ad
146 .sp .6
147 .RS 4n
148 \fB/home\fR (can be overwritten in \fB/etc/default/useradd\fR file)
149 .RE

```

```

151 .sp
152 .ne 2
153 .na
154 \fBskel_dir\fR
155 .ad
156 .sp .6
157 .RS 4n
158 \fB/etc/skel\fR
159 .RE

```

```

161 .sp
162 .ne 2
163 .na
164 \fBshell\fR
165 .ad
166 .sp .6
167 .RS 4n
168 \fB/bin/sh\fR
169 .RE

```

```

171 .sp
172 .ne 2
173 .na
174 \fBinactive\fR
175 .ad
176 .sp .6
177 .RS 4n
178 \fB0\fR
179 .RE

```

```

181 .sp
182 .ne 2
183 .na
184 \fBexpire\fR
185 .ad
186 .sp .6
187 .RS 4n
188 null
189 .RE

```

```

191 .sp

```

```

192 .ne 2
193 .na
194 \fBauths\fR
195 .ad
196 .sp .6
197 .RS 4n
198 null
199 .RE

```

```

201 .sp
202 .ne 2
203 .na
204 \fBprofiles\fR
205 .ad
206 .sp .6
207 .RS 4n
208 null
209 .RE

```

```

211 .sp
212 .ne 2
213 .na
214 \fBproj\fR
215 .ad
216 .sp .6
217 .RS 4n
218 \fB3\fR
219 .RE

```

```

221 .sp
222 .ne 2
223 .na
224 \fBprojname\fR
225 .ad
226 .sp .6
227 .RS 4n
228 \fBdefault\fR
229 .RE

```

```

231 .sp
232 .ne 2
233 .na
234 \fBkey=value (pairs defined in \fBuser_attr(4)\fR
235 .ad
236 .sp .6
237 .RS 4n
238 not present
239 .RE

```

```

241 .sp
242 .ne 2
243 .na
244 \fBroles\fR
245 .ad
246 .sp .6
247 .RS 4n
248 null
249 .RE

```

```

251 If \fBusr/sadm/defadduser\fR file does not exist or doesn't set basedir
252 (for example, on first \fBuseradd -D\fR invocation), default value for
253 \fBbase_dir\fR is determined by \fBHOME_PARENT\fR parameter in
254 \fB/etc/default/useradd\fR file.
255 .RE

```

```

257 .sp

```

```

258 .ne 2
259 .na
260 \fB\fB-e\fR \fIexpire\fR\fR
261 .ad
262 .sp .6
263 .RS 4n
264 Specify the expiration date for a login. After this date, no user will be able
265 to access this login. The expire option argument is a date entered using one of
266 the date formats included in the template file \fB/etc/datemsk\fR. See
267 \fBgetdate\fR(3C).
268 .sp
269 If the date format that you choose includes spaces, it must be quoted. For
270 example, you can enter \fB10/6/90\fR or \fBOctober 6, 1990\fR. A null value
271 (\fB" "\fR) defeats the status of the expired date. This option is useful for
272 creating temporary logins.
273 .RE

275 .sp
276 .ne 2
277 .na
278 \fB\fB-f\fR \fIinactive\fR\fR
279 .ad
280 .sp .6
281 .RS 4n
282 The maximum number of days allowed between uses of a login ID before that
283 \fBID\fR is declared invalid. Normal values are positive integers. A value of
284 \fB0\fR defeats the status.
285 .RE

287 .sp
288 .ne 2
289 .na
290 \fB\fB-g\fR \fIgroup\fR\fR
291 .ad
292 .sp .6
293 .RS 4n
294 An existing group's integer \fBID\fR or character-string name. Without the
295 \fB-D\fR option, it defines the new user's primary group membership and
296 defaults to the default group. You can reset this default value by invoking
297 \fBuseradd\fR \fB-D\fR \fB-g\fR \fIgroup\fR. GIDs 0-99 are reserved for
298 allocation by the Solaris Operating System.
299 .RE

301 .sp
302 .ne 2
303 .na
304 \fB\fB-G\fR \fIgroup\fR\fR
305 .ad
306 .sp .6
307 .RS 4n
308 An existing group's integer \fBID\fR or character-string name. It defines the
309 new user's supplementary group membership. Duplicates between \fIgroup\fR with
310 the \fB-g\fR and \fB-G\fR options are ignored. No more than \fBNGROUPS_MAX\fR
311 groups can be specified. GIDs 0-99 are reserved for allocation by the Solaris
312 Operating System.
313 .RE

315 .sp
316 .ne 2
317 .na
318 \fB\fB-K\fR \fIkey=value\fR\fR
319 .ad
320 .sp .6
321 .RS 4n
322 A \fIkey=value\fR pair to add to the user's attributes. Multiple \fB-K\fR
323 options may be used to add multiple \fIkey=value\fR pairs. The generic \fB-K\fR

```

```

324 option with the appropriate key may be used instead of the specific implied key
325 options (\fB-A\fR, \fB-P\fR, \fB-R\fR, \fB-p\fR). See \fBuser_attr\fR(4) for a
326 list of valid \fIkey=value\fR pairs. The "type" key is not a valid key for this
327 option. Keys may not be repeated.
328 .RE

330 .sp
331 .ne 2
332 .na
333 \fB\fB-k\fR \fIskel_dir\fR\fR
334 .ad
335 .sp .6
336 .RS 4n
337 A directory that contains skeleton information (such as \fB&.profile\fR) that
338 can be copied into a new user's home directory. This directory must already
339 exist. The system provides the \fB/etc/skel\fR directory that can be used for
340 this purpose.
341 .RE

343 .sp
344 .ne 2
345 .na
346 \fB\fB-m\fR\fR [\fB-z|-Z\fR]
347 .ad
348 .sp .6
349 .RS 4n
350 Create the new user's home directory if it does not already exist. If the
351 directory already exists, it must have read, write, and execute permissions by
352 \fIgroup\fR, where \fIgroup\fR is the user's primary group.
353 .sp
354 If the parent directory of the user's home directory is located on a separate
355 \fBZFS\fR file system and the \fB/etc/default/useradd\fR file contains the
356 parameter \fBMANAGE_ZFS\fR set to the value \fBYES\fR, a new \fBZFS\fR
357 file system will be created for the user.
358 .sp
359 If the \fB-z\fR option is specified, \fBuseradd\fR will always try to create
360 a new file system for the home directory.
361 .sp
362 If the \fB-Z\fR option is specified, a new file system will never be created.
363 .RE

365 .sp
366 .ne 2
367 .na
368 \fB\fB-o\fR\fR
369 .ad
370 .sp .6
371 .RS 4n
372 This option allows a \fBUID\fR to be duplicated (non-unique).
373 .RE

375 .sp
376 .ne 2
377 .na
378 \fB\fB-P\fR \fIprofile\fR\fR
379 .ad
380 .sp .6
381 .RS 4n
382 One or more comma-separated execution profiles defined in \fBprof_attr\fR(4).
383 .RE

385 .sp
386 .ne 2
387 .na
388 \fB\fB-p\fR \fIprojname\fR\fR
389 .ad

```

```

390 .sp .6
391 .RS 4n
392 Name of the project with which the added user is associated. See the
393 \fIprojname\fR field as defined in \fBproject\fR(4).
394 .RE

396 .sp
397 .ne 2
398 .na
399 \fB\fB-R\fR \fIrole\fR
400 .ad
401 .sp .6
402 .RS 4n
403 One or more comma-separated execution profiles defined in \fBUser_attr\fR(4).
404 Roles cannot be assigned to other roles.
405 .RE

407 .sp
408 .ne 2
409 .na
410 \fB\fB-s\fR \fIshell\fR
411 .ad
412 .sp .6
413 .RS 4n
414 Full pathname of the program used as the user's shell on login. It defaults to
415 an empty field causing the system to use \fB/bin/sh\fR as the default. The
416 value of \fIshell\fR must be a valid executable file.
417 .RE

419 .sp
420 .ne 2
421 .na
422 \fB\fB-u\fR \fIuid\fR
423 .ad
424 .sp .6
425 .RS 4n
426 The \fBUID\fR of the new user. This \fBUID\fR must be a non-negative decimal
427 integer below \fBMAXUID\fR as defined in \fB<sys/param.h>\fR. The \fBUID\fR
428 defaults to the next available (unique) number above the highest number
429 currently assigned. For example, if \fBUID\fRs 100, 105, and 200 are assigned,
430 the next default \fBUID\fR number will be 201. \fBUID\fRs \fB0\fR-\fB99\fR are
431 reserved for allocation by the Solaris Operating System.
432 .RE

434 .SH FILES
435 .LP
436 \fB/etc/default/useradd\fR
437 .sp
438 .LP
439 \fB/etc/datemsk\fR
440 .sp
441 .LP
442 \fB/etc/passwd\fR
443 .sp
444 .LP
445 \fB/etc/shadow\fR
446 .sp
447 .LP
448 \fB/etc/group\fR
449 .sp
450 .LP
451 \fB/etc/skel\fR
452 .sp
453 .LP
454 \fB/etc/user_attr\fR
455 .sp

```

```

456 .LP
457 \fB/usr/include/limits.h\fR
458 .sp
459 .LP
460 \fB/usr/sadm/defadduser\fR
461 \fB/etc/user_attr\fR
461 .SH ATTRIBUTES
462 .LP
463 See \fBAttributes\fR(5) for descriptions of the following attributes:
464 .sp

466 .sp
467 .TS
468 box;
469 c | c
470 l | l .
471 ATTRIBUTE TYPE ATTRIBUTE VALUE
472 _
473 Interface Stability Committed
474 .TE

476 .SH SEE ALSO
477 .LP
478 \fBpasswd\fR(1), \fBprofiles\fR(1), \fBroles\fR(1), \fBUsers\fR(1B),
479 \fBgroupadd\fR(1M), \fBgroupdel\fR(1M), \fBgroupmod\fR(1M), \fBgrpck\fR(1M),
480 \fBlogins\fR(1M), \fBpwck\fR(1M), \fBuserdel\fR(1M), \fBusermod\fR(1M),
481 \fBgetdate\fR(3C), \fBauth_attr\fR(4), \fBpasswd\fR(4), \fBprof_attr\fR(4),
482 \fBproject\fR(4), \fBUser_attr\fR(4), \fBAttributes\fR(5), \fBzfs\fR(1M)
483 .SH DIAGNOSTICS
484 .LP
485 In case of an error, \fBuseradd\fR prints an error message and exits with a
486 non-zero status.
487 .sp
488 .LP
489 The following indicates that \fBlogin\fR specified is already in use:
490 .sp
491 .in +2
492 .nf
493 UX: useradd: ERROR: login is already in use. Choose another.
494 .fi
495 .in -2
496 .sp

498 .sp
499 .LP
500 The following indicates that the \fIuid\fR specified with the \fB-u\fR option
501 is not unique:
502 .sp
503 .in +2
504 .nf
505 UX: useradd: ERROR: uid \fIuid\fR is already in use. Choose another.
506 .fi
507 .in -2
508 .sp

510 .sp
511 .LP
512 The following indicates that the \fIgroup\fR specified with the \fB-g\fR option
513 is already in use:
514 .sp
515 .in +2
516 .nf
517 UX: useradd: ERROR: group \fIgroup\fR does not exist. Choose another.
518 .fi
519 .in -2
520 .sp

```

```
522 .sp
523 .LP
524 The following indicates that the \fIuid\fR specified with the \fB-u\fR option
525 is in the range of reserved \fBUID\fRs (from \fB0\fR-\fB99\fR):
526 .sp
527 .in +2
528 .nf
529 UX: useradd: WARNING: uid \fIuid\fR is reserved.
530 .fi
531 .in -2
532 .sp

534 .sp
535 .LP
536 The following indicates that the \fIuid\fR specified with the \fB-u\fR option
537 exceeds \fBMAXUID\fR as defined in \fB<sys/param.h>\fR:
538 .sp
539 .in +2
540 .nf
541 UX: useradd: ERROR: uid \fIuid\fR is too big. Choose another.
542 .fi
543 .in -2
544 .sp

546 .sp
547 .LP
548 The following indicates that the \fB/etc/passwd\fR or \fB/etc/shadow\fR files
549 do not exist:
550 .sp
551 .in +2
552 .nf
553 UX: useradd: ERROR: Cannot update system files - login cannot be created.
554 .fi
555 .in -2
556 .sp

558 .SH NOTES
559 .LP
560 The \fBuseradd\fR utility adds definitions to only the local \fB/etc/group\fR,
561 \fB/etc/passwd\fR, \fB/etc/passwd\fR, \fB/etc/shadow\fR, \fB/etc/project\fR, and
562 \fB/etc/user_attr\fR files. If a network name service such as \fBNIS\fR or
563 \fBNIS+\fR is being used to supplement the local \fB/etc/passwd\fR file with
564 additional entries, \fBuseradd\fR cannot change information supplied by the
565 network name service. However \fBuseradd\fR will verify the uniqueness of the
566 user name (or role) and user id and the existence of any group names specified
567 against the external name service.
```