NAME

getuserattr, getusernam, setuserattr, enduserattr – get user_attr entry

SYNOPSIS

cc [*flag* ...] *file* ... **–lsecdb –lsocket –lnsl –lintl** [*library* ...]

#include <user_attr.h>

userattr_t *getuserattr(void);

userattr_t *getusernam(const char **name***);**

userattr_t *getuseruid(uid_t *uid***);**

void free_userattr(userattr_t **userattr***);**

void setuserattr(void);

void enduserattr(void);

DESCRIPTION

The **getuserattr( )**, **getusernam( )**, and **getuseruid( )** functions each return a **user_attr**(4) entry. Entries can come from any of the sources specified in the **nsswitch.conf**(4) file. The **getuserattr( )** function enumerates **user_attr** entries. The **getusernam( )** function searches for a **user_attr** entry with a given user name *name*. The **getuseruid( )** function searches for a **user_attr** entry with a given user id *uid*. Successive calls to these functions return either successive **user_attr** entries or **NULL**.

The **free_userattr( )** function call releases memory allocated by the **getusernam( )** and **getuserattr( )** functions.

The internal representation of a **user_attr** entry is a **userattr_t** structure defined in **<user_attr.h>** with the following members:

```
char  *name;/* name of the user */
char  *qualifier;/* reserved for future use */
char  *res1;/* reserved for future use */
char  *res2;/* reserved for future use */
kva_t*attr;/* list of attributes */
```

The **setuserattr( )** function "rewinds" to the beginning of the enumeration of **user_attr** entries. Calls to **getusernam( )** may leave the enumeration in an indeterminate state, so **setuserattr( )** should be called before the first call to **getuserattr( )**.

The **enduserattr( )** function may be called to indicate that **user_attr** processing is complete; the library may then close any open **user_attr** file, deallocate any internal storage, and so forth.

RETURN VALUES

The **getuserattr( )** function returns a pointer to a **userattr_t** if it successfully enumerates an entry; otherwise it returns **NULL**, indicating the end of the enumeration.

The **getusernam( )** function returns a pointer to a **userattr_t** if it successfully locates the requested entry; otherwise it returns **NULL**.

Individual attributes may be referenced in the **attr** structure by calling the **kva_match**(3) function.

CAVEATS

Because the list of legal keys is likely to expand, code must be written to ignore unknown key-value pairs without error.

FILES

/etc/user_attr	stores extended user attributes
/etc/nsswitch.conf	stores configuration file lookup information for the name server switch

ATTRIBUTES

See **attributes**(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe.

SEE ALSO

getauthattr(3), **getexecattr**(3), **getprofattr**(3), and **user_attr**(4)

NOTES

getuserattr( ) and **getusernam( )** both allocate memory for the pointers they return. This memory should be de-allocated with the **free_userattr()** call.

Programs that use the interfaces described in this manual page cannot be linked statically, since the implementations of these functions employ dynamic loading and linking of shared objects at run time. Note that these interfaces are reentrant even though they do not use the _r suffix naming convention.