z/OS Documentation Updates for APAR PH57896

Table of Contents

XL C/C++ Runtime Library Reference

Chapter 2. Header files

pwd.h — Access the user database through password structure

The pwd.h header file declares functions that access the user database through a password structure. The header file also defines the passwd structure.

_POSIX_SOURCE

getpwnam() getpwuid()

_XOPEN_SOURCE_EXTENDED 1

endpwent() getpwent() setpwent()

_XPLATFORM_SOURCE 1

getpwent_r()

Chapter 3. Library functions

endpwent() — User database functions

Standards

Standards / Extensions	C or C++	Dependencies
XPG4.2 Single UNIX Specification, Version 3 z/OS UNIX	both	

Format

```
#define _XOPEN_SOURCE_EXTENDED 1

#include <pwd.h>

void endpwent(void);

struct passwd *getpwent(void);

void setpwent(void);

#define _XPLATFORM_SOURCE 1

#include <pwd.h>

int getpwent_r(struct passwd *pwbuf, char *buf, size_t buflen, struct passwd **pwbufp);
```

General Description

The getpwent() function returns a pointer to the broken-out fields of a line in the user database, mapped by the passwd structure defined in the <pwd.h> header file. The password structure is saved in a thread-specific storage that is reused in each call by the same thread. Repeated calls to getpwent() return a pointer to the next passwd structure in the database, until End Of File (EOF), at which point a NULL pointer is returned. setpwent() interrupts this sequential search and rewinds the user database to the beginning, such that the next getpwent() returns a pointer to the first passwd structure. Use of setpwent() is optional after an End Of File (EOF), as the next getpwent() after end of file again returns a pointer to the first passwd structure. endpwent() is optionally used to close the user database when searching is complete.

The setpwent() function effectively rewinds the user database to allow repeated searches.

The endpwent() function may be called to close the user database when processing is complete.

The getpwent_r() function is the reentrant version of getpwent(). The password structure returned as pointer pwbufp is saved in the caller supplied buffer pwbuf. The string fields associated with the password structure are saved in the caller supplied buffer buf of size buflen.

Returned value
When first called, getpwent() returns a pointer to the next passwd structure in the user database. Upon subsequent calls it returns a pointer to a passwd structure, or it returns a NULL pointer on either End Of File (EOF) or an error. The return value may point to static data that is overwritten by each call. Unlike getpwent(), getpwent_r() returns zero with *pwbufp set to the pointer of the password structure in the successful cases, and returns a non-zero value with *pwbufp set to NULL in failed cases.

There are no documented errno values for endpwent() and setpwent(). For getpwent() and getpwent_r(), if unsuccessful, errno is set to one of the following values:

Error Code

Description

EINVAL

getpwent_r() only: Invalid parameter supplied.

EMVSSAF2ERR

The system authorization facility (SAF) or RACF Get GMAP service had an error.

EMVSSAFEXTRERR

The SAF or RACF RACROUTE EXTRACT service had an error.

ENOENT

getpwent_r() only: No more entries.

ERANGE

getpwent_r() only: Insufficient buffer space supplied. Try again with larger buffer.

Related Information

- “pwd.h — Access user database through password structure” on page 61
- “getgrent() — Get group database entry” on page 672
- “getgrgid() — Access the group database by ID” on page 672
- “getgrnam() — Access the group database by name” on page 675
- “getlogin() — Get the user login name” on page 695
- “getpwent() — Get user database entry” on page 722
- “getpwent_r() — Get user database entry reentrantly” on page XXX

getpwent_r() — Get user database entry reentrantly

The information for this function is included in “endpwent() — User database functions” on page XXX.

Language Environment Vendor Interfaces

Chapter 1. Common interfaces and conventions

Language Environment common anchor area

Append the description of CEECAACEL4VEC4 to ***Table 19 Common anchor area (CAA) field descriptions***:

Table 19. Common anchor area (CAA) field descriptions (continued)					
Offsets		Type	Len	Name	Description
Decimal	Hex				
1032	(408)	Address	4	CEECAACEL4VEC4	Address of 4th C-RTL library vector.

Append the description of CEECAACEL4VEC4 to ***Table 21. Common anchor area (CAA) cross reference***:

Table 21. Common anchor area (CAA) cross reference (continued)			
Name	Hex Offset	Hex Value	Level
CEECAACEL4VEC4	408		3

Append the description of CEECAACEL4VEC4 to ***CAA fields***:

CEECAACEL4VEC4

Address of 4th C-RTL library vector.