NAME

safe_strto_id_list, safe_strto_uid_list, safe_strto_gid_list – Parse a string for ids and store them in the safe_id_range_list

SYNOPSIS

#include "safe_id_range_list.h"

struct *safe_id_range_list***;**
typedef struct safe_id_range_list *safe_id_range_list***;**

void safe_strto_id_list(safe_id_range_list *list***, const char *** *value***, const char **** *endptr***);**
void safe_strto_uid_list(safe_id_range_list *list***, const char *** *value***, const char **** *endptr***);**
void safe_strto_gid_list(safe_id_range_list *list***, const char *** *value***, const char **** *endptr***);**

DESCRIPTION

These functions will add values to the *list* by parsing the *value* string. **safe_strto_id_list only allows numeric values, while safe_strto_uid_list** and **safe_strto_gid_list** allow non-numeric values that are converted to user and group ids respectively.

The functions **safe_parse_id_list**(3), **safe_parse_uid_list**(3), and **safe_parse_gid_list**(3) are simpler to use alternatives.

Parameters

list The safe_id_range_list to add values to. Must be non-**NULL**.

value The range list string to parse. Must be non-**NULL**.

endptr If non-**NULL**, the value pointed to by this parameter will be updated to point to the location just past the last character of *value* parsed.

Range List Syntax

The syntax of a range list string is a set of ranges each separated by colons. Each range is either a single id, or a hyphen ('-') separated set of ids (the first must be less than or equal to the second and the second value may be a valid user or group name depending the function used. If a user or group name is used as the first part of a range the name must be separated from the hyphen by whitespace as a hyphen is a legitimate character in a name.

Parsing stops when an expected punctuation mark ('-' or ':') is not found. This is not an error and *endptr must be checked to detect this condition. A grammar for the range list is as follows:

<RANGE_LIST>
:= <RANGE>
:= <RANGE> <whitespace>* ':' <RANGE_LIST>

<RANGE>
:= <ID>
:= <ID> '-' <ID>
:= <ID> '-' <whitespace>* '*'

<ID> := <whitespace>* <number>
:= <whitespace>* <text_name>

Example Range Lists

3 Adds the value 3 to the list.

0:5:100-103
Adds the values 0, 5, 100, 101, 102, and 103 to the list.

root: user1 - user2 : 1000-*
When parsed with **safe_strto_uid_list**, adds the values 0, 500, 501 and 1000 and greater to the list assuming the user names root, user1 and user2 have the values 0, 500 and 501 respectively. The space after user1 is mandatory.

RETURN VALUE

No return value, but *errno* is set to 0 on success and an appropriate value on error. Also *endptr* if non-**NULL**, will be set to the location in value just past the last value parsed.

ERRORS

EINVAL

list or *value* was **NULL**. An invalid number or non-exist user or group name was encountered. The first element of a range was larger than the second. Or a value consisting entirely of white-space was encountered.

ERANGE

A numeric value was encountered that was too large.

ENOMEM

A memory allocation failed.

AUTHOR

James A. Kupsch

SEE ALSO

safe_init_id_range_list(3) **safe_destroy_id_range_list**(3) **safe_parse_id_list**(3) **safe_parse_uid_list**(3) **safe_parse_gid_list**(3)

http://research.cs.wisc.edu/mist/safefile