Rocky Enterprise Linux 9.2 Manual Pages on command 'envz_get.3'

$ man envz_get.3

ENVZ_ADD(3)　　　　　Linux Programmer's Manual　　　　　ENVZ_ADD(3)

NAME

　　envz_add, envz_entry, envz_get, envz_merge, envz_remove, envz_strip -

　　environment string support

SYNOPSIS

　　#include <envz.h>

　　error_t envz_add(char **envz, size_t *envz_len,

　　　　　　const char *name, const char *value);

　　char *envz_entry(const char *envz, size_t envz_len, const char *name);

　　char *envz_get(const char *envz, size_t envz_len, const char *name);

　　error_t envz_merge(char **envz, size_t *envz_len,

　　　　　　　const char *envz2, size_t envz2_len, int override);

　　void envz_remove(char **envz, size_t *envz_len, const char *name);

　　void envz_strip(char **envz, size_t *envz_len);

DESCRIPTION

　　These functions are glibc-specific.

　　An argz vector is a pointer to a character buffer together with a

　　length, see argz_add(3).   An envz vector is a special argz vector,

namely one where the strings have the form "name=value". Everything after the first '=' is considered to be the value. If there is no '=', the value is taken to be NULL. (While the value in case of a trailing '=' is the empty string "".)

These functions are for handling envz vectors.

envz_add() adds the string "name=value" (in case value is non-NULL) or "name" (in case value is NULL) to the envz vector (*envz, *envz_len) and updates *envz and *envz_len. If an entry with the same name ex‐ isted, it is removed.

envz_entry() looks for name in the envz vector (envz, envz_len) and re‐ turns the entry if found, or NULL if not.

envz_get() looks for name in the envz vector (envz, envz_len) and re‐ turns the value if found, or NULL if not. (Note that the value can also be NULL, namely when there is an entry for name without '=' sign.)

envz_merge() adds each entry in envz2 to *envz, as if with envz_add(). If override is true, then values in envz2 will supersede those with the same name in *envz, otherwise not.

envz_remove() removes the entry for name from (*envz, *envz_len) if there was one.

envz_strip() removes all entries with value NULL.

RETURN VALUE

All envz functions that do memory allocation have a return type of er‐ ror_t (an integer type), and return 0 for success, and ENOMEM if an al‐ location error occurs.

ATTRIBUTES

For an explanation of the terms used in this section, see at‐ tributes(7).

Interface	Attribute	Value
envz_add(), envz_entry(), envz_get(), envz_merge(), envz_remove(), envz_strip()	Thread safety	MT-Safe

???

CONFORMING TO

These functions are a GNU extension.

EXAMPLES

```c
#include <stdio.h>

#include <stdlib.h>

#include <envz.h>

int

main(int argc, char *argv[], char *envp[])

{

    int e_len = 0;

    char *str;

    for (int i = 0; envp[i] != NULL; i++)

        e_len += strlen(envp[i]) + 1;

    str = envz_entry(*envp, e_len, "HOME");

    printf("%s\n", str);

    str = envz_get(*envp, e_len, "HOME");

    printf("%s\n", str);

    exit(EXIT_SUCCESS);

}
```

SEE ALSO

argz_add(3)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A
description of the project, information about reporting bugs, and the
latest   version   of   this   page,   can   be   found   at
https://www.kernel.org/doc/man-pages/.