NAME

curl_easy_escape - URL encodes the given string

SYNOPSIS

```
#include <curl/curl.h>

char *curl_easy_escape(CURL *curl, const char *string, int length);
```

DESCRIPTION

This function converts the given input *string* to a URL encoded string and returns that as a new allocated string. All input characters that are not a-z, A-Z, 0-9, ’-’, ’.’, ’_’ or ’~’ are converted to their "URL escaped" version (**%NN** where **NN** is a two-digit hexadecimal number).

If *length* is set to 0 (zero), *curl_easy_escape(3)* uses strlen() on the input *string* to find out the size. This function does not accept input strings longer than **CURL_MAX_INPUT_LENGTH** (8 MB).

Since 7.82.0, the **curl** parameter is ignored. Prior to that there was per-handle character conversion support for some old operating systems such as TPF, but it was otherwise ignored.

You must *curl_free(3)* the returned string when you are done with it.

ENCODING

libcurl is typically not aware of, nor does it care about, character encodings. *curl_easy_escape(3)* encodes the data byte-by-byte into the URL encoded version without knowledge or care for what particular character encoding the application or the receiving server may assume that the data uses.

The caller of *curl_easy_escape(3)* must make sure that the data passed in to the function is encoded correctly.

EXAMPLE

```
int main(void)
{
 CURL *curl = curl_easy_init();
 if(curl) {
  char *output = curl_easy_escape(curl, "data to convert", 15);
  if(output) {
   printf("Encoded: %s\n", output);
   curl_free(output);
  }
  curl_easy_cleanup(curl);
```

```
  }
}
```

AVAILABILITY

Added in 7.15.4 and replaces the old *curl_escape(3)* function.

RETURN VALUE

A pointer to a null-terminated string or NULL if it failed.

SEE ALSO

curl_easy_unescape(3), **curl_free**(3)