

NAME

CURLOPT_FTP_USE_EPRT – use EPRT for FTP

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_FTP_USE_EPRT, long enabled);
```

DESCRIPTION

Pass a long. If the value is 1, it tells curl to use the EPRT command when doing active FTP downloads (which is enabled by *CURLOPT_FTPPORT*(3)). Using EPRT means that libcurl first attempts to use EPRT before using PORT, but if you pass zero to this option, it avoids using EPRT, only plain PORT.

The EPRT command is a slightly newer addition to the FTP protocol than PORT and is the preferred command to use since it enables IPv6 to be used. Old FTP servers might not support it, which is why libcurl has a fallback mechanism. Sometimes that fallback is not enough and then this option might come handy.

If the server is an IPv6 host, this option has no effect as EPRT is necessary then.

DEFAULT**PROTOCOLS**

This functionality affects ftp only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "ftp://example.com/file.txt");

        /* contact us back, aka "active" FTP */
        curl_easy_setopt(curl, CURLOPT_FTPPORT, "-");

        /* FTP the way the neanderthals did it */
        curl_easy_setopt(curl, CURLOPT_FTP_USE_EPRT, 0L);

        result = curl_easy_perform(curl);

        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.10.5

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors*(3).

SEE ALSO

CURLOPT_FTPPORT(3), **CURLOPT_FTP_USE_EPSV**(3)