

NAME

CURLOPT_CONV_FROM_NETWORK_FUNCTION – convert data from network to host encoding

SYNOPSIS

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

```
CURLcode conv_callback(char *ptr, size_t length);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_CONV_FROM_NETWORK_FUNCTION,
                           conv_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

Applies to non-ASCII platforms. *curl_version_info(3)* returns the **CURL_VERSION_CONV** feature bit set if this option is provided.

The data to be converted is in a buffer pointed to by the *ptr* parameter. The amount of data to convert is indicated by the *length* parameter. The converted data overlays the input data in the buffer pointed to by the *ptr* parameter. *CURLE_OK* must be returned upon successful conversion. A *CURLcode* return value defined by *curl.h*, such as *CURLE_CONV_FAILED*, should be returned if an error was encountered.

CURLOPT_CONV_FROM_NETWORK_FUNCTION(3) converts to host encoding from the network encoding. It is used when commands or ASCII data are received over the network.

If you set a callback pointer to *NULL*, or do not set it at all, the built-in libcurl iconv functions are used. If *HAVE_ICONV* was not defined when libcurl was built, and no callback has been established, the conversion returns the **CURLE_CONV_REQD** error code.

If **HAVE_ICONV** is defined, **CURL_ICONV_CODESET_OF_HOST** must also be defined. For example:

```
#define CURL_ICONV_CODESET_OF_HOST "IBM-1047"
```

The iconv code in libcurl defaults the network and UTF8 codeset names as follows:

```
#define CURL_ICONV_CODESET_OF_NETWORK "ISO8859-1"
```

```
#define CURL_ICONV_CODESET_FOR_UTF8 "UTF-8"
```

You need to override these definitions if they are different on your system.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
static CURLcode my_conv_from_ascii_to_ebcdic(char *buffer, size_t length)
{
    int rc = 0;

    /* in-place convert 'buffer' from ASCII to EBCDIC */

    if(rc == 0) {
        /* success */
    }
}
```

```
    return CURLE_OK;
}
else {
    return CURLE_CONV_FAILED;
}
}

int main(void)
{
    CURL *curl = curl_easy_init();

    /* use platform-specific functions for codeset conversions */
    curl_easy_setopt(curl, CURLLOPT_CONV_FROM_NETWORK_FUNCTION,
                     my_conv_from_ascii_to_ebcdic);
}
```

DEPRECATED

Not available and deprecated since 7.82.0.

Available only if **CURL_DOES_CONVERSIONS** was defined when libcurl was built.

AVAILABILITY

Added in curl 7.15.4

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

CURLLOPT_CONV_FROM_UTF8_FUNCTION(3), **CURLLOPT_CONV_TO_NETWORK_FUNCTION(3)**