NAME

CURLOPT_INTERLEAVEFUNCTION − callback for RTSP interleaved data

SYNOPSIS

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

size_t interleave_callback(void *ptr, size_t size, size_t nmemb,
                           void *userdata);

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_INTERLEAVEFUNCTION,
                          interleave_callback);
```

DESCRIPTION

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

This callback function gets called by libcurl as soon as it has received interleaved RTP data. This function gets called for each $ block and therefore contains exactly one upper−layer protocol unit (e.g. one RTP packet). curl writes the interleaved header as well as the included data for each call. The first byte is always an ASCII dollar sign. The dollar sign is followed by a one byte channel identifier and then a 2 byte integer length in network byte order. See RFC 2326 Section 10.12 for more information on how RTP interleaving behaves. If unset or set to NULL, curl uses the default write function.

Interleaved RTP poses some challenges for the client application. Since the stream data is sharing the RTSP control connection, it is critical to service the RTP in a timely fashion. If the RTP data is not handled quickly, subsequent response processing may become unreasonably delayed and the connection may close. The application may use *CURL_RTSPREQ_RECEIVE* to service RTP data when no requests are desired. If the application makes a request, (e.g. *CURL_RTSPREQ_PAUSE*) then the response handler processes any pending RTP data before marking the request as finished.

The *CURLOPT_INTERLEAVEDATA(3)* is passed in the *userdata* argument in the callback.

Your callback should return the number of bytes actually taken care of. If that amount differs from the amount passed to your callback function, it signals an error condition to the library. This causes the transfer to abort and the libcurl function used returns *CURLE_WRITE_ERROR*.

You can also abort the transfer by returning CURL_WRITEFUNC_ERROR. (7.87.0)

DEFAULT

NULL, the interleave data is then passed to the regular write function: *CURLOPT_WRITEFUNCTION(3)*.

PROTOCOLS

This functionality affects rtsp only

EXAMPLE

```
struct local {
  void *custom;
};

static size_t rtp_write(void *ptr, size_t size, size_t nmemb, void *userp)
{
  struct local *l = userp;
  printf("our ptr: %p\n", l->custom);
  /* take care of the packet in 'ptr', then return... */
  return size * nmemb;
}

int main(void)
```

```
{
 struct local rtp_data;
 CURL *curl = curl_easy_init();
 if(curl) {
  curl_easy_setopt(curl, CURLOPT_INTERLEAVEFUNCTION, rtp_write);
  curl_easy_setopt(curl, CURLOPT_INTERLEAVEDATA, &rtp_data);
 }
}
```

AVAILABILITY

Added in curl 7.20.0

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_INTERLEAVEDATA(3), **CURLOPT_RTSP_REQUEST**(3)