

**NAME**

CURLOPT\_RTSP\_REQUEST – specify RTSP request

**SYNOPSIS**

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_RTSP_REQUEST, long request);
```

**DESCRIPTION**

Tell libcurl what kind of RTSP request to make. Pass one of the following RTSP enum values as a long in the *request* argument. Unless noted otherwise, commands require the Session ID to be initialized.

**CURL\_RTSPREQ\_OPTIONS**

Used to retrieve the available methods of the server. The application is responsible for parsing and obeying the response. **(The session ID is not needed for this method.)**

**CURL\_RTSPREQ\_DESCRIBE**

Used to get the low level description of a stream. The application should note what formats it understands in the 'Accept:' header. Unless set manually, libcurl will automatically fill in 'Accept: application/sdp'. Time-condition headers will be added to Describe requests if the *CURLOPT\_TIMECONDITION(3)* option is active. **(The session ID is not needed for this method)**

**CURL\_RTSPREQ\_ANNOUNCE**

When sent by a client, this method changes the description of the session. For example, if a client is using the server to record a meeting, the client can use Announce to inform the server of all the meta-information about the session. ANNOUNCE acts like a HTTP PUT or POST just like *CURL\_RTSPREQ\_SET\_PARAMETER*

**CURL\_RTSPREQ\_SETUP**

Setup is used to initialize the transport layer for the session. The application must set the desired Transport options for a session by using the *CURLOPT\_RTSP\_TRANSPORT(3)* option prior to calling setup. If no session ID is currently set with *CURLOPT\_RTSP\_SESSION\_ID(3)*, libcurl will extract and use the session ID in the response to this request. **(The session ID is not needed for this method).**

**CURL\_RTSPREQ\_PLAY**

Send a Play command to the server. Use the *CURLOPT\_RANGE(3)* option to modify the playback time (e.g. 'npt=10-15').

**CURL\_RTSPREQ\_PAUSE**

Send a Pause command to the server. Use the *CURLOPT\_RANGE(3)* option with a single value to indicate when the stream should be halted. (e.g. npt='25')

**CURL\_RTSPREQ\_TEARDOWN**

This command terminates an RTSP session. Simply closing a connection does not terminate the RTSP session since it is valid to control an RTSP session over different connections.

**CURL\_RTSPREQ\_GET\_PARAMETER**

Retrieve a parameter from the server. By default, libcurl will automatically include a *Content-Type: text/parameters* header on all non-empty requests unless a custom one is set. GET\_PARAMETER acts just like a HTTP PUT or POST (see *CURL\_RTSPREQ\_SET\_PARAMETER*). Applications wishing to send a heartbeat message (e.g. in the presence of a server-specified timeout) should send use an empty GET\_PARAMETER request.

**CURL\_RTSPREQ\_SET\_PARAMETER**

Set a parameter on the server. By default, libcurl will automatically include a *Content-Type: text/parameters* header unless a custom one is set. The interaction with SET\_PARAMETER is much like a HTTP PUT or POST. An application may either use *CURLOPT\_UPLOAD(3)* with *CURLOPT\_READDATA(3)* like a HTTP PUT, or it may use *CURLOPT\_POSTFIELDS(3)* like a HTTP POST. No chunked transfers are allowed, so the application must set the *CURLOPT\_INFILESIZE(3)* in the former and *CURLOPT\_POSTFIELDSIZE(3)* in the latter. Also,

there is no use of multi-part POSTs within RTSP.

#### CURL\_RTSPREQ\_RECORD

Used to tell the server to record a session. Use the *CURLOPT\_RANGE(3)* option to modify the record time.

#### CURL\_RTSPREQ\_RECEIVE

This is a special request because it does not send any data to the server. The application may call this function in order to receive interleaved RTP data. It will return after processing one read buffer of data in order to give the application a chance to run.

### **DEFAULT PROTOCOLS EXAMPLE**

TODO

### **AVAILABILITY**

Added in 7.20.0

### **RETURN VALUE**

Returns CURLE\_OK if the option is supported, and CURLE\_UNKNOWN\_OPTION if not.

### **SEE ALSO**

**CURLOPT\_RTSP\_SESSION\_ID(3), CURLOPT\_RTSP\_STREAM\_URI(3),**