NAME

librtmp - RTMPDump Real-Time Messaging Protocol API

LIBRARY

RTMPDump RTMP (librtmp, -lrtmp)

SYNOPSIS

#include <librtmp/rtmp.h>

DESCRIPTION

The Real-Time Messaging Protocol (RTMP) is used for streaming multimedia content across a TCP/IP network. This API provides most client functions and a few server functions needed to support RTMP, RTMP tunneled in HTTP (RTMPT), encrypted RTMP (RTMPE), RTMP over SSL/TLS (RTMPS) and tunneled variants of these encrypted types (RTMPTE, RTMPTS). The basic RTMP specification has been published by Adobe but this API was reverse-engineered without use of the Adobe specification. As such, it may deviate from any published specifications but it usually duplicates the actual behavior of the original Adobe clients.

The RTMPDump software package includes a basic client utility program in **rtmpdump**(1), some sample servers, and a library used to provide programmatic access to the RTMP protocol. This man page gives an overview of the RTMP library routines. These routines are found in the -lrtmp library. Many other routines are also available, but they are not documented yet.

The basic interaction is as follows. A session handle is created using **RTMP_Alloc**() and initialized using **RTMP_Init**(). All session parameters are provided using **RTMP_SetupURL**(). The network connection is established using **RTMP_Connect**(), and then the RTMP session is established using **RTMP_ConnectStream**(). The stream is read using **RTMP_Read**(). A client can publish a stream by calling **RTMP_EnableWrite**() before the **RTMP_Connect**() call, and then using **RTMP_Write**() after the session is established. While a stream is playing it may be paused and unpaused using **RTMP_Pause**(). The stream playback position can be moved using **RTMP_Seek**(). When **RTMP_Read**() returns 0 bytes, the stream is complete and may be closed using **RTMP_Close**(). The session handle is freed using **RTMP_Free**().

All data is transferred using FLV format. The basic session requires an RTMP URL. The RTMP URL format is of the form

rtmp[t][e|s]://hostname[:port][/app[/playpath]]

Plain rtmp, as well as tunneled and encrypted sessions are supported.

Additional options may be specified by appending space-separated key=value pairs to the URL.

Special characters in values may need to be escaped to prevent misinterpretation by the option parser. The escape encoding uses a backslash followed by two hexadecimal digits representing the ASCII value of the character. E.g., spaces must be escaped as **\20** and backslashes must be escaped as **\5c**.

OPTIONS

Network Parameters

These options define how to connect to the media server.

socks=*host:port*
: Use the specified SOCKS4 proxy.

Connection Parameters

These options define the content of the RTMP Connect request packet. If correct values are not provided, the media server will reject the connection attempt.

app=*name*
: Name of application to connect to on the RTMP server. Overrides the app in the RTMP URL. Sometimes the librtmp URL parser cannot determine the app name automatically, so it must be given explicitly using this option.

tcUrl=*url*
: URL of the target stream. Defaults to rtmp[t][e|s]://host[:port]/app.

pageUrl=*url*
: URL of the web page in which the media was embedded. By default no value will be sent.

swfUrl=*url*
: URL of the SWF player for the media. By default no value will be sent.

flashVer=*version*
: Version of the Flash plugin used to run the SWF player. The default is "LNX 10,0,32,18".

conn=*type:data*
: Append arbitrary AMF data to the Connect message. The type must be B for Boolean, N for number, S for string, O for object, or Z for null. For Booleans the data must be either 0 or 1 for FALSE or TRUE, respectively. Likewise for Objects the data must be 0 or 1 to end or begin an object, respectively. Data items in subobjects may be named, by prefixing the type with 'N' and specifying the name before the value, e.g. NB:myFlag:1. This option may be used multiple times to construct arbitrary AMF sequences. E.g.
  conn=B:1 conn=S:authMe conn=O:1 conn=NN:code:1.23 conn=NS:flag:ok conn=O:0

Session Parameters

These options take effect after the Connect request has succeeded.

playpath=*path*

Overrides the playpath parsed from the RTMP URL. Sometimes the rtmpdump URL parser cannot determine the correct playpath automatically, so it must be given explicitly using this option.

playlist=*0|1*

If the value is 1 or TRUE, issue a set_playlist command before sending the play command. The playlist will just contain the current playpath. If the value is 0 or FALSE, the set_playlist command will not be sent. The default is FALSE.

live=*0|1*

Specify that the media is a live stream. No resuming or seeking in live streams is possible.

subscribe=*path*

Name of live stream to subscribe to. Defaults to *playpath*.

start=*num*

Start at *num* seconds into the stream. Not valid for live streams.

stop=*num*

Stop at *num* seconds into the stream.

buffer=*num*

Set buffer time to *num* milliseconds. The default is 30000.

timeout=*num*

Timeout the session after *num* seconds without receiving any data from the server. The default is 120.

Security Parameters

These options handle additional authentication requests from the server.

token=*key*

Key for SecureToken response, used if the server requires SecureToken authentication.

jtv=*JSON*

JSON token used by legacy Justin.tv servers. Invokes NetStream.Authenticate.UsherToken

swfVfy=*0|1*

If the value is 1 or TRUE, the SWF player is retrieved from the specified *swfUrl* for performing SWF Verification. The SWF hash and size (used in the verification step) are computed automatically. Also the SWF information is cached in a *.swfinfo* file in the user's home directory, so that it doesn't need to be retrieved and recalculated every time. The .swfinfo file records the SWF URL, the time it was fetched, the modification timestamp of the SWF file, its size, and its hash. By default, the cached info will be used for 30 days before re-checking.

swfAge=*days*

Specify how many days to use the cached SWF info before re-checking. Use 0 to always check the SWF URL. Note that if the check shows that the SWF file has the same modification timestamp as before, it will not be retrieved again.

EXAMPLES

An example character string suitable for use with **RTMP_SetupURL**():

"rtmp://flashserver:1935/ondemand/thefile swfUrl=http://flashserver/player.swf swfVfy=1"

ENVIRONMENT

HOME

The value of $**HOME** is used as the location for the *.swfinfo* file.

FILES

$HOME/.swfinfo

Cache of SWF Verification information

SEE ALSO

rtmpdump(1), **rtmpgw**(8)

AUTHORS

Andrej Stepanchuk, Howard Chu, The Flvstreamer Team
<http://rtmpdump.mplayerhq.hu>