



This repository Search

[Explore](#) [Gist](#) [Blog](#) [Help](#)

itpp16

arut / **nginx-rtmp-module**[Watch](#) ▾

293

[★ Star](#)

1,579

[Fork](#)

452

NGINX-based Media Streaming Server <http://nginx-rtmp.blogspot.com>[943](#) commits[150](#) branches[105](#) releases[10](#) contributorsbranch: **master** ▾**nginx-rtmp-module** / +

enabled static relay cleanup to fix static_relay+auto_push combination

**arut** authored on Sep 23, 2014latest commit [7a35372e30](#)

dash	Explicitly signal HE-AAC in DASH manifest	6 months ago
doc	added documentation link	2 years ago
hls	Update ngx_rtmp_mpegts.c	5 months ago
test	removed trailing space	a year ago
AUTHORS	improved frame timing & added meta files	3 years ago
LICENSE	updated year	a year ago
README.md	Use same path in the comments and in the code	a year ago
config	added proxy protocol support	11 months ago
ngx_rtmp.c	added queue init	5 months ago
ngx_rtmp.h	fixed compilation	5 months ago
ngx_rtmp_access_module.c	removed trailing spaces	a year ago
ngx_rtmp_amf.c	improved Envivio fix: now key length can be unfinalized with object	a year ago
ngx_rtmp_amf.h	updated copyright line	a year ago

[Code](#)[Issues](#)

220
























[Pull Requests](#)



















24

[Wiki](#)[Pulse](#)[Graphs](#)

HTTPS clone URL

<https://github.com/arut/n>You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). ⓘ[Clone in Desktop](#)[Download ZIP](#)

 ngx_rtmp_auto_push_module.c	renamed variable from sun to saun to fix SmarOS compilation	a year ago
 ngx_rtmp_bandwidth.c	removed trailing spaces	a year ago
 ngx_rtmp_bandwidth.h	added audio and video bw to stat	a year ago
 ngx_rtmp_bitop.c	implemented avc sps parser & improved dash & stats	a year ago
 ngx_rtmp_bitop.h	implemented avc sps parser & improved dash & stats	a year ago
 ngx_rtmp_cmd_module.c	implemented play2 time continuation	a year ago
 ngx_rtmp_cmd_module.h	shared fill args func	a year ago
 ngx_rtmp_codec_module.c	fixed windows compilation	a year ago
 ngx_rtmp_codec_module.h	added aac-he & aac-hev2 support	a year ago
 ngx_rtmp_control_module.c	fixed errors in control output; now returning http 206 instead of emp...	a year ago
 ngx_rtmp_core_module.c	added proxy protocol support	11 months ago
 ngx_rtmp_eval.c	removed trailing spaces	a year ago
 ngx_rtmp_eval.h	merged exec pull feature	a year ago
 ngx_rtmp_exec_module.c	added filename, basename & dirname to exec_record_done	a year ago
 ngx_rtmp_flv_module.c	implemented default buflen setting	a year ago
 ngx_rtmp_handler.c	reset byte counter at 0xf0000000 to fix publishing from fmle	a year ago
 ngx_rtmp_handshake.c	removed trailing spaces	a year ago
 ngx_rtmp_init.c	added queue init	5 months ago
 ngx_rtmp_limit_module.c	removed trailing spaces	a year ago
 ngx_rtmp_live_module.c	reverted changes in live module	a year ago
 ngx_rtmp_live_module.h	reverted changes in live module	a year ago
 ngx_rtmp_log_module.c	fixed issue with multiple access_logs	5 months ago
 ngx_rtmp_mp4_module.c	fixed skipping first key frame in mp4 streamer	10 months ago

 ngx_rtmp_netcall_module.c	added clientid to notifications & control	a year ago
 ngx_rtmp_netcall_module.h	removed trailing spaces	a year ago
 ngx_rtmp_notify_module.c	added epoch argument to on_connect	a year ago
 ngx_rtmp_play_module.c	fixed compilation	5 months ago
 ngx_rtmp_play_module.h	removed trailing spaces	a year ago
 ngx_rtmp_proxy_protocol.c	fixed compilation with old nginx	11 months ago
 ngx_rtmp_proxy_protocol.h	added proxy protocol support	11 months ago
 ngx_rtmp_receive.c	removed trailing spaces	a year ago
 ngx_rtmp_record_module.c	added timestamp cut in recorder to prevent negative (big unsigned) ti...	a year ago
 ngx_rtmp_record_module.h	recorder now writes correct flv track mask	a year ago
 ngx_rtmp_relay_module.c	enabled static relay cleanup to fix static_relay+auto_push combination	5 months ago
 ngx_rtmp_relay_module.h	removed trailing spaces	a year ago
 ngx_rtmp_send.c	removed trailing spaces	a year ago
 ngx_rtmp_shared.c	removed trailing spaces	a year ago
 ngx_rtmp_stat_module.c	fixed formatting error in stat module	a year ago
 ngx_rtmp_streams.h	updated copyright line	a year ago
 ngx_rtmp_version.h	version bump	11 months ago
 stat.xsl	moved bytes/bw block to the right of stat page	a year ago

README.md

NGINX-based Media Streaming Server

nginx-rtmp-module

Project blog

<http://nginx-rtmp.blogspot.com>

Wiki manual

<https://github.com/arut/nginx-rtmp-module/wiki/Directives>

Google group

<https://groups.google.com/group/nginx-rtmp>

<https://groups.google.com/group/nginx-rtmp-ru> (Russian)

Donation page (Paypal etc)

<http://arut.github.com/nginx-rtmp-module/>

Features

- RTMP/HLS/MPEG-DASH live streaming
- RTMP Video on demand FLV/MP4, playing from local filesystem or HTTP
- Stream relay support for distributed streaming: push & pull models
- Recording streams in multiple FLVs
- H264/AAC support

- Online transcoding with FFmpeg
- HTTP callbacks (publish/play/record/update etc)
- Running external programs on certain events (exec)
- HTTP control module for recording audio/video and dropping clients
- Advanced buffering techniques to keep memory allocations at a minimum level for faster streaming and low memory footprint
- Proved to work with Wirecast, FMS, Wowza, JWPlayer, FlowPlayer, StrobeMediaPlayback, ffmpeg, avconv, rtmpdump, flvstreamer and many more
- Statistics in XML/XSL in machine- & human- readable form
- Linux/FreeBSD/MacOS/Windows

Build

cd to NGINX source directory & run this:

```
./configure --add-module=/path/to/nginx-rtmp-module  
make  
make install
```

Several versions of nginx (1.3.14 - 1.5.0) require `http_ssl_module` to be added as well:

```
./configure --add-module=/path/to/nginx-rtmp-module --with-http_ssl_module
```

For building debug version of nginx add `--with-debug`

```
./configure --add-module=/path/to/nginx/rtmp-module --with-debug
```

[Read more about debug log](#)

Windows limitations

Windows support is limited. These features are not supported

- execs
- static pulls
- auto_push

RTMP URL format

```
rtmp://rtmp.example.com/app[/name]
```

app - should match one of application {} blocks in config

name - interpreted by each application can be empty

Multi-worker live streaming

Module supports multi-worker live streaming through automatic stream pushing to nginx workers. This option is toggled with rtmp_auto_push directive.

Example nginx.conf

```
rtmp {
```

```
server {

    listen 1935;

    chunk_size 4000;

    # TV mode: one publisher, many subscribers
    application mytv {

        # enable live streaming
        live on;

        # record first 1K of stream
        record all;
        record_path /tmp/av;
        record_max_size 1K;

        # append current timestamp to each flv
        record_unique on;

        # publish only from localhost
        allow publish 127.0.0.1;
        deny publish all;

        #allow play all;
    }

    # Transcoding (ffmpeg needed)
    application big {
        live on;

        # On every published stream run this command (ffmpeg)
        # with substitutions: $app/${app}, $name/${name} for application & stream name.
        #
        # This ffmpeg call receives stream from this application &
        # reduces the resolution down to 32x32. The stream is the published to
        # 'small' application (see below) under the same name.
        #
    }
}
```

```
# ffmpeg can do anything with the stream like video/audio
# transcoding, resizing, altering container/codecs etc
#
# Multiple exec lines can be specified.

exec ffmpeg -re -i rtmp://localhost:1935/$app/$name -vcodec flv -acodec copy -s 32x32
    -f flv rtmp://localhost:1935/small/${name};
}

application small {
    live on;
    # Video with reduced resolution comes here from ffmpeg
}

application webcam {
    live on;

    # Stream from local webcam
    exec_static ffmpeg -f video4linux2 -i /dev/video0 -c:v libx264 -an
        -f flv rtmp://localhost:1935/webcam/mystream;
}

application mypush {
    live on;

    # Every stream published here
    # is automatically pushed to
    # these two machines
    push rtmp1.example.com;
    push rtmp2.example.com:1934;
}

application mypull {
    live on;

    # Pull all streams from remote machine
    # and play locally
    pull rtmp://rtmp3.example.com pageUrl=www.example.com/index.html;
```



```
}

application mystaticpull {
    live on;

    # Static pull is started at nginx start
    pull rtmp://rtmp4.example.com pageUrl=www.example.com/index.html name=mystream static
}

# video on demand
application vod {
    play /var/flvs;
}

application vod2 {
    play /var/mp4s;
}

# Many publishers, many subscribers
# no checks, no recording
application videochat {

    live on;

    # The following notifications receive all
    # the session variables as well as
    # particular call arguments in HTTP POST
    # request

    # Make HTTP request & use HTTP retcode
    # to decide whether to allow publishing
    # from this connection or not
    on_publish http://localhost:8080/publish;

    # Same with playing
    on_play http://localhost:8080/play;

    # Publish/play end (repeats on disconnect)
```

```
    on_done http://localhost:8080/done;

    # All above mentioned notifications receive
    # standard connect() arguments as well as
    # play/publish ones. If any arguments are sent
    # with GET-style syntax to play & publish
    # these are also included.
    # Example URL:
    #   rtmp://localhost/myapp/mystream?a=b&c=d

    # record 10 video keyframes (no audio) every 2 minutes
    record keyframes;
    record_path /tmp/vc;
    record_max_frames 10;
    record_interval 2m;

    # Async notify about an flv recorded
    on_record_done http://localhost:8080/record_done;

}

# HLS

# For HLS to work please create a directory in tmpfs (/tmp/hls here)
# for the fragments. The directory contents is served via HTTP (see
# http{} section in config)
#
# Incoming stream must be in H264/AAC. For iPhones use baseline H264
# profile (see ffmpeg example).
# This example creates RTMP stream from movie ready for HLS:
#
# ffmpeg -loglevel verbose -re -i movie.avi -vcodec libx264
#       -vprofile baseline -acodec libmp3lame -ar 44100 -ac 1
#       -f flv rtmp://localhost:1935/hls/movie
#
# If you need to transcode live stream use 'exec' feature.
#
```

```
application hls {
    live on;
    hls on;
    hls_path /tmp/hls;
}

# MPEG-DASH is similar to HLS

application dash {
    live on;
    dash on;
    dash_path /tmp/dash;
}
}

# HTTP can be used for accessing RTMP stats
http {

    server {

        listen      8080;

        # This URL provides RTMP statistics in XML
        location /stat {
            rtmp_stat all;

            # Use this stylesheet to view XML as web page
            # in browser
            rtmp_stat_stylesheet stat.xml;
        }

        location /stat.xml {
            # XML stylesheet to view RTMP stats.
            # Copy stat.xml wherever you want
            # and put the full directory path here
            root /path/to/stat.xml;
        }
    }
}
```

```
location /hls {
    # Serve HLS fragments
    types {
        application/vnd.apple.mpegurl m3u8;
        video/mp2t ts;
    }
    root /tmp;
    add_header Cache-Control no-cache;
}

location /dash {
    # Serve DASH fragments
    root /tmp;
    add_header Cache-Control no-cache;
}
}
```

Multi-worker streaming example

```
rtmp_auto_push on;

rtmp {
    server {
        listen 1935;

        application mytv {
            live on;
        }
    }
}
```

