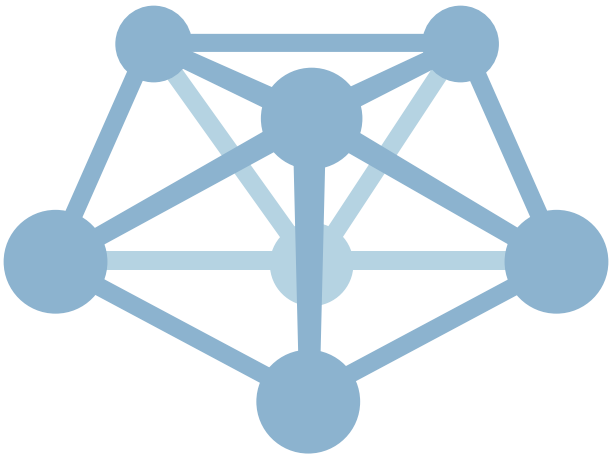
MistServer documentation

DDVTech

August 19, 2015



Live Streaming

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

Contents

1 Live streaming **3**
1.1 Understanding live streaming . . . 3
1.2 When and Why should I use a live stream? . . . 3
1.3 Setting up your live stream . . . 3
1.3.1 Generic instructions . . . 3
1.3.2 MistServer Settings . . . 4
1.3.2.1 Setting up your protocols . . . 4
1.3.2.2 Setting up your live buffer **MistServer Pro version only* . . . 4
1.3.2.3 Setting up your stream . . . 4
1.3.2.4 MistServer Source settings for live stream examples . . . 4
1.3.3 Live RTMP ingest . . . 4
1.3.3.1 RTMP live source application settings . . . 5
1.3.4 Examples using various live source applications . . . 5
1.3.4.1 ffmpeg push examples . . . 5
1.3.4.2 FMLE push example . . . 6
1.3.4.3 Elemental push example . . . 6
1.3.4.4 VMix push example . . . 6
1.3.4.5 OBS push example . . . 7
1.3.4.6 Xsplit push example . . . 7
1.4 Multibitrate streaming **MistServer Pro version only* . . . 7
1.4.1 What is Multibitrate streaming? . . . 7
1.4.2 When would I use Multibitrate streaming? . . . 7
1.4.3 How do I use Multibitrate streaming? . . . 8
1.5 Wildcard streaming **MistServer Pro version only* . . . 8
1.5.1 What is Wildcard streaming? . . . 8
1.5.2 When would I use Wildcard streaming? . . . 8
1.5.3 How to use Wildcard streaming . . . 8
1.5.3.1 Example of using Wildcard streaming . . . 8

Appendix A Codec support **9**

Appendix B Supported inputs and outputs and specification **9**

Appendix C MistIns list **9**

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

1 Live streaming

The following document is automatically generated by the MistServer system to include documentation/instructions available at the above mentioned date.

The following document will contain `identifiers` and their *values*, styled as done here. Identifiers are names for things, while values is what they contain. For example the identifier `url` can contain the value *http://localhost/*.

1.1 Understanding live streaming

Live streaming differs from video on demand (VoD) on one simple aspect. Video on demand (VoD) streams have a set beginning and end and do not change over time. Live streams are in principle infinite but there's an ever changing availability window. Live streams can be recorded turning them into VoD streams.

1.2 When and Why should I use a live stream?

Live streaming is really useful when every viewer needs to be able to see the same content at the same time. You could compare it to watching the news on TV compared to watching something on YouTube.
While both methods will work to reach your audience it's best to decide between the two methods on one simple criterion: the content. If you have a single non-changing piece of content you'd probably want to use VoD, if you have content that is ongoing or changes with certain events outside of your control.
Some good examples of live streams would be: Event streaming (contests, conferences, shows, etc.) and newsreports.
Some good examples of VoD streams would be: movies, podcasts, TV series.

Of course, using one method doesn't automatically exclude you from the other, it's possible to combine methods. You can record a live stream and put it available as VoD while it's still running which is often done at huge events or conferences. Just as you can make a VoD "live" by live streaming it as is done with nearly every TV series. Those techniques are for more advanced users and will be beyond the scope of this document however.

So if you're going to set up a stream, make sure you consider the benefits and drawbacks of both live and VoD for your stream to optimize your viewer experience.

1.3 Setting up your live stream

If you'd like to set up a single live stream please follow the generic instructions. If you want to set up a flexible live stream which can accept and stream multiple sources at the same time please see our Pro only feature: Wildcard.
You'll start with loggin into your MistServer by going to *http://MistServerIP:4242* and logging in using the login information you've filled in during first boot, if you haven't created a login try to log in using anything and you'll be able to create a login account.

1.3.1 Generic instructions

The hardest part of live streaming is getting the host, port, application and stream correct on both MistServer and your device pushing towards MistServer. It's important to remember both MistServer and your pushing application need the address of each other, just remember they need to communicate with eachother and it will start making more sense.

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

1.3.2 MistServer Settings

Regardless of how you'll push towards MistServer the MistServer settings will stay the same. You can set up a live stream in two steps. First you need to set the RTMP port, then you need to create a live stream. Viewers of live streams will automatically connect to the point as close to live as possible for the protocol, some protocols can be more live than others.

1.3.2.1 Setting up your protocols

If you have activated every protocol (the standard option) you can skip this step. If you haven't activated any protocols you can open them at protocols within your MistServer Management Interface. The standard port for HTTP is 8080, RTMP uses 1935. Protocols are responsible of how MistServer can communicate with other devices, adding or editing the protocols tells MistServer how to behave. Currently MistServer only supports RTMP pushing, so make sure you've one enabled if you want to attempt live streaming.

1.3.2.2 Setting up your live buffer

MistServer Pro version only

Currently MistServer will automatically connect new viewers as close to live as possible. Pro users however have the option to connect new viewers further back. You can do this with the `Starting position in live buffer`. The higher the number the more live your stream is, 1000 is as live as possible while 0 is as far away from live as possible.

1.3.2.3 Setting up your stream

To create a live stream, go to streams and create a new stream. Set up a `stream name` and use *push://ip.address* as a source. Do remember that you'll probably have to fill in the `stream name` for each device pushing towards MistServer, so we recommend something short and easy to remember.

`Stream name`
This is the name used by both MistServer and any device pushing to MistServer to connect correctly. MistServer handles a limit of 100 characters for your stream name, names are not case sensitive and will decline special characters.

`Source`
Here you can tell MistServer whether the stream is a live stream and from what locations it should accept a push and set a password (Pro only). The source is fully compatible with both ip addresses and host names, please make sure the ip addresses or host names you fill in are correct.

1.3.2.4 MistServer Source settings for live stream examples

push://ip.address	Live stream accepting pushes only from the set location.
push://hostname	Live stream accepting pushes only from the corresponding hostname.
push://	Live stream accepting pushes from any location.
*push://ip.address@password**	Live stream accepting pushes from the set location using the set password.
*push://@password**	Live stream accepting pushes from any location using the set password.

**MistServer Pro version only*

1.3.3 Live RTMP ingest

Currently MistServer only supports RTMP ingest, luckily RTMP ingest suffices for nearly any use case.
A RTMP url exists of a `host`, `port`, `application` and `stream`. This looks like `rtmp://host:port/application/stream`.
Most software will let you change all of these, though some software will call them differently.

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

Please note that MistServer has an unique RTMP implementation when it comes to using a *"/"* in the `stream`. Anything following a *"/"* in the `stream` when pushing to MistServer will be dropped. While we do not allow special characters in our `stream` and most users will not be bothered by this it'll allow for a better multibitrate support for certain applications.

1.3.3.1 RTMP live source application settings

First remember that MistServer and your live source application need to communicate. So your application will need the information to connect to MistServer, proper use of `host`, `port`, `application` and `stream` will decide your succes.

`host`	This will be your MistServer address.
`port`	If you're using the standard port you can omit this from your url, otherwise you'll need to use speci
`application`	This is used as another identifier, if you're using MistServer Pro you can use it as a password. If you
`stream`	This needs to correspond with the stream name you've set in MistServer

Some examples of `host`, `port`, `application` and `stream`, as *live* is the most commonly used `application` name we'll use that as well:

Pushing to MistServer and MistServer runs on the same host, standard RTMP port, no password.
rtmp://localhost/live/StreamName

Pushing to MistServer, standard RTMP port, no password.
rtmp://MistServer.ip.address/live/StreamName

Pushing to MistServer, custom RTMP port, no password.
rtmp://MistServer.ip.address:CustomPort/live/StreamName

Pushing to MistServer with password.
rtmp://MistServer.ip.address/password/StreamName

1.3.4 Examples using various live source applications

Incase the generic instructions aren't clear enough we've add some of more used live source applications and what the identifiers are called below.

1.3.4.1 ffmpeg push examples

ffmpeg syntax file as live push.
ffmpeg -re -i FileName.Extension -acodec copy -vcodec copy -f flv rtmp://MistServer.ip.address/live/StreamName

ffmpeg syntax camera as live push
ffmpeg -i CameraURL -acodec copy -vcodec copy -f flv rtmp://MistServer.ip.address/live/StreamName

ffmpeg syntax camera as live push, force H264/AAC
ffmpeg -i CameraURL -acodec aac -strict -2 -vcodec h264 -f flv rtmp://MistServer.ip.address/live/StreamName

ffmpeg syntax camera as live push RTSP over TCP to prevent package loss
ffmpeg -rtsp_transport tcp -i CameraURL -acodec copy -vcodec copy -f flv rtmp://MistServer.ip.address/live/StreamName

ffmpeg syntax input as live push with password*
ffmpeg -i CameraURL -acodec copy -vcodec copy -f flv rtmp://MistServer.ip.address/Password/StreamName
**MistServer Pro version only*

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

1.3.4.2 FMLE push example

FMLE push same machine, standard RTMP port, no password
`FMS Url`: *rtmp://localhost:1935/live* `Stream`: *StreamName*

FMLE push other machine, standard RTMP port, no password
`FMS Url`: *rtmp://MistServer.ip.address:1935/live* `Stream`: *StreamName*

FMLE push other machine, custom RTMP port, no password
`FMS Url`: *rtmp://MistServer.ip.address:CustomPort/live* `Stream`: *StreamName*

FMLE push other machine*, standard RTMP port, with password*
`FMS Url`: *rtmp://MistServer.ip.address:1935/password* `Stream`: *StreamName*
**MistServer Pro version only*

FMLE Multibitrate exception
****MistServer Pro version only***

FMLE has an exception for Multibitrate pushing as FMLE pushes multibitrate streams with the tracks specified in the push urls. MistServer excpects them under the same name however. To solve this problem you'll have to use *"/%i"* instead of *"%i"* at the end of your `FMS URL`. MistServer will drop the different names and detect the multiple streams as a multibitrate push.

1.3.4.3 Elemental push example

Elemental push other machine, standard RTMP port, no password
`RTMP Endpoint`: *rtmp://MistServer.ip.address:1935/live* `Stream Name`: *StreamName*

Elemental push other machine, custom RTMP port, no password
`RTMP Endpoint`: *rtmp://MistServer.ip.address:CustomPort/live* `Stream Name`: *StreamName*

FMLE push other machine*, standard RTMP port, with password*
`RTMP Endpoint`: *rtmp://MistServer.ip.address:1935/password* `Stream Name`: *StreamName*
**MistServer Pro version only*

Elemental Multibitrate exception
****MistServer Pro version only***

Elemental has an exception for Multibitrate pushing as you have to combine multiple pushes towards the same stream using Elemental. In order to do this you'll have to create 2 or more streams and have each push push towards a different `Stream Name`. The first stream should be named "*StreamName/1*", the second should be called "*StreamName/2*", repeat this for any following number of streams.

1.3.4.4 VMix push example

VMix push same machine, standard RTMP port, no password
`Url`: *rtmp://localhost:1935/live* `Stream Key`: *StreamName*

VMix push other machine, standard RTMP port, no password
`Url`: *rtmp://MistServer.ip.address:1935/live* `Stream Key`: *StreamName*

VMix push other machine, custom RTMP port, no password
`Url`: *rtmp://MistServer.ip.address:CustomPort/live* `Stream Key`: *StreamName*

VMix push other machine, standard RTMP port, with password*
`Url`: *rtmp://MistServer.ip.address:1935/password* `Stream key`: *StreamName*
**MistServer Pro version only*

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

1.3.4.5 OBS push example

You can find the stream settings at settings, Broadcast settings.

OBS push same machine, standard RTMP port, no password
`FMS url`: *rtmp://localhost:1935/live* `Stream Key`: *StreamName*

OBS push other machine, standard RTMP port, no password
`FMS url`: *rtmp://MistServer.ip.address:1935/live* `Stream Key`: *StreamName*

OBS push other machine, custom RTMP port, no password
`FMS url`: *rtmp://MistServer.ip.address:CustomPort/live* `Stream Key`: *StreamName*

OBS push other machine, standard RTMP port, with password*
`FMS url`: *rtmp://MistServer.ip.address:1935/password* `Stream key`: *StreamName*
**MistServer Pro version only*

1.3.4.6 Xsplit push example

You can add the MistServer to push at Broadcast, add channel, Custom RTMP. You'll need to set up the RTMP Url and Stream Name.

Xsplit push same machine, standard RTMP port, no password
`RTMP Url`: *rtmp://localhost:1935/live* `Stream name`: *StreamName*

Xsplit push other machine, standard RTMP port, no password
`RTMP Url`: *rtmp://MistServer.ip.address:1935/live* `Stream name`: *StreamName*

Xsplit push other machine, custom RTMP port, no password
`RTMP Url`: *rtmp://MistServer.ip.address:CustomPort/live* `Stream name`: *StreamName*

Xsplit push other machine, standard RTMP port, with password*
`RTMP Url`: *rtmp://MistServer.ip.address:1935/password* `Stream name`: *StreamName*
**MistServer Pro version only*

1.4 Multibitrate streaming

MistServer Pro version only

1.4.1 What is Multibitrate streaming?

Multibitrate streams are simply streams with multiple video and/or audio tracks so viewers/players are able to select the tracks best fitting for their experience.

1.4.2 When would I use Multibitrate streaming?

Multibitrate streaming has multiple uses, it's most common use is simply using it as a way to reach all possible viewers with just one stream. Due to device/hardware restrictions not every viewer can handle the same quality of stream, multibitrate helps you reach all of the viewers by making every quality you're pushing available.

Other uses are to add subtitles or different languages through multiple tracks, you can even add commentary on one track and have the viewers select the track they prefer to listen to. You can even stream two different cameras at the same time and have viewers select their preferred view, although this will most likely just confuse the viewers.

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

1.4.3 How do I use Multibitrate streaming?

Multibitrate streaming is quite simple when using MistServer, simply use the same application to stream multiple qualities or even an entirely different set of tracks through another application. As long as you're pushing to MistServer using the same `stream name` it'll automatically put it available as a multibitrate track, provided it's supported by the protocols/codecs available to MistServer.

Some applications have an unique way to handle Multibitrate, please check push examples per application for exceptions.

1.5 Wildcard streaming

MistServer Pro version only

1.5.1 What is Wildcard streaming?

Wildcard streaming is MistServers solution to easily make a system that supports multiple live streams by accepting an everchangable `stream`. With Wildcard streaming the `stream` indentifier isn't set to a specific value. Which means you can send an indefinite amount of streams towards your Wildcard stream making them all available under the `stream` value you decide when you push to MistServer.

1.5.2 When would I use Wildcard streaming?

You'd love Wildcard streaming if you have to set up multiple live streams or if you live stream a lot and want each stream to be available under a new name.

If you host a big event it can be quite a bother to set up a new live stream for each part you're going to record. With Wildcard streaming you can set up a single stream and push towards it under different names instantly making the streams available under names you choose. Another use would be if you live stream a lot and want each stream to be available under a new name.

1.5.3 How to use Wildcard streaming

Every live stream has the potential to be a Wildcard stream, Wildcard streaming can be activated by adding *+Wildcardname* to any set `stream` value. As you'd have to add the first `stream` value to every Wildcard stream we'd advice an short and easy to remember name. Every Wildcard stream follows the same rules as any other live stream, the only difference is that Wildcard streams will only exist for as long as there's a connection open. Wildcard streams will only shut down as soon as there's no connection for at least 20 seconds, which means it will only close 20 seconds after you're done adding new content and your viewers are finished watching your stream. Another benefit is that Wildcards names are only restricted by the amount of characters. The amount of characters cannot exceed 100, but you're free to use any type of special character you'd want.

1.5.3.1 Example of using Wildcard streaming

In this example we've added a stream called *live* to MistServer. We could use this as a normal live stream simply by using the `stream` indentifier *live* to push to MistServer. To enable wildcard however we'd have to add a *+Wildcardname*. Lets say we'd like to stream under the wildcard name "August 19, 2015" we'd have to push towards MistServer with the following `stream` value: *live+August 19, 2015*.
The stream *live+August 19, 2015* will now be available as a live stream for any of the supported protocols.

Phone: +1 408 677 5212
Phone: +31 10 8080 619
E-mail: info@ddvtech.com

Appendix A Codec support

	H264	HEVC	AAC	MP3	Flash*	Theora	Vorbis	SRT	WebVTT	Metadata
DASH	✓	✓	✓	✗	-	✗	✗	-	-	-
HDS (Adobe)	✓	-	✓	✓	✓	✗	✗	-	-	-
HLS (Apple)	✓	✓	✓	-	-	✗	✗	-	✗	-
HSS (Silverlight)	✓	✓	✓	✓	✗	✗	-	-	-	-
FLV	✓	-	✓	✓	✓	✗	✗	-	-	-
MP3	✓	-	✓	✓	✓	✗	✗	-	-	-
MP4	✓	✓	✓	✓	✗	✗	✗	-	-	-
RTMP	✓	-	✓	✓	✓	✗	✗	-	-	✓
RTSP	✓	-	✓	✓	✗	✗	✗	-	-	-
TS	✓	✓	✓	✓	✗	✗	✗	✗	✗	-
OGG	✗	✗	✗	✗	✗	✓	✓	✗	✗	-
HTTP JSON	-	-	-	-	-	✗	✗	✓	✓	✓
HTTP SRT	-	-	-	-	-	✗	✗	✓	✓	-

**The flash codecs are VP6, JPEG, H.263, Screen Video, 1/2, PCM, Nelly, G711 and Speex*

Appendix B Supported inputs and outputs and specification

	VoD input	Live input	VoD output	Live output	Seeking	Multibitrate
DASH	✗	✗	✓	✓	✓	✓
HDS	✗	✗	✓	✓	✓	✓
HLS	✗	✗	✓	✓	✓	✓
HSS	✓	✗	✓	✓	✓	✓
FLV	✓	✗	✓	✓	✓	-
MP3	✓	✗	✓	✓	✗	-
MP4	✓	✗	✓	✓	✓	-
RTMP	-	✓	✓	✓	✓	SMIL
RTSP	-	✗	✓	✓	✓	-
TS	✓	✓	✓	✓	-	✓
OGG	✓	✗	✓	✓	✓	✗

Appendix C MistIns list

MistInput	Association
MistInBuffer	Live streaming
MistInDTSC	DTSC / DTSH files
MistInFLV	FLV file input
MistInFolder	Folder support
MistInISMV	ISMV file input
MistInMP3	MP3 file input
MistInMP4	MP4 file input
MistInOGG	OGG file input
MistInTS	TS file / stream input