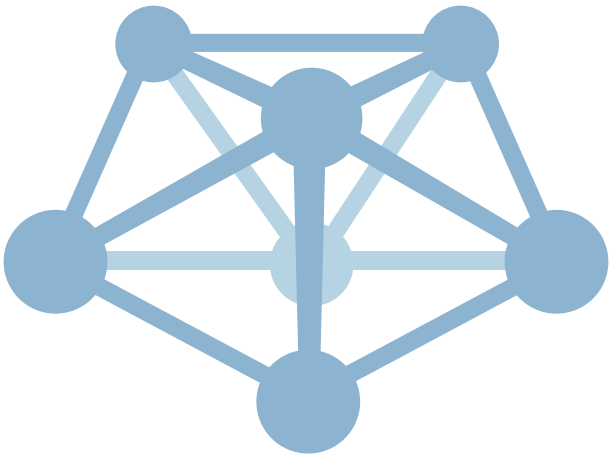
MistServer documentation

DDVTech

August 19, 2015



MistServer Manual

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

Contents

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

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

1 Installing MistServer

1.1 Do you need to install MistServer to run it?

Shortly put: No, you do not need to install MistServer in order to run it. You can download the binaries at our product page and boot MistServer by running the MistController(.exe). While MistServer doesn't require to be run as administrator it can help certain issues on certain devices, we recommend running it as administrator.

Compiling is the closest you can get to installing MistServer on your system is possible and it will ensure your MistServer install is optimised for your system.

1.2 Installing MistServer on your device

1.2.1 Compiling MistServer

To compile MistServer you'll need the source, you can find it at our github page. We do not provide the source for MistServer Pro, we do however provide custom builds optimised for your system. Please contact us if you're interested in a custom build. Compiling MistServer itself can be done by a simple command:

```
cmake <path_to_Mist_source> && make && sudo make install
```

This will prepare, build and install MistServer on your system. If you'd like to build MistServer in the same folder simply use "." as the path.

1.2.2 MistServer on your Raspberry Pi

MistServer can run on your Raspberry Pi, we provide entire images of an Arch Linux ARM with MistServer preinstalled. You can find the download at our download page. Simply install the downloaded image to a SD card and boot your Raspberry Pi. If you're unsure how to install the image please review the documentation at raspberrypi.com They've got a really detailed explanation available for you.

The standard log in to your Raspberry Pi will always be the following:

Login:	root
Password:	root

We recommend changing these if you'll use your Raspberry Pi in an open environment.

1.2.3 MistServer on your Virtual Private Server (VPS)

Installing MistServer on your VPS can be done quite quickly as MistServer itself has no dependencies. As MistServer runs the best on Linux we recommend a version of Linux as your VPS Operating System. While you can compile MistServer on your VPS the same way as you could on your own devices most VPS will be without a compiler. Thus the binaries are usually easier to install and see if MistServer works.

1.2.3.1 Installing MistServer on your VPS

If you're using MistServer open source please check our downloads page to check the <URL> for the latest MistServer open source release. If you're using MistServer Pro you can use the <URL> provided to you in the notification email or available on the *MistServer account* page. You might need to download and install packages in order to download and unpack MistServer. You can download and unpack MistServer to the correct folder using the following command:

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

```
wget <URL> -O mist.tgz
tar -xf mist.tgz -C /usr/bin
```

It's possible you need to install the `wget` package in order to download MistServer. Please look up the proper package and install method for your Operating System. You've now installed MistServer, to run MistServer you can use the following command:

```
MistController
```

Feel free to add parameter like `--Log file.log` to save the MistServer logs as well, use `MistServer -h` for a complete list.

1.2.4 MistServer as a service

MistServer can be used as a service, you can do so by using the `mistserver.init` or `mistserver.service` found in the *extra* folder. When you run this script MistServer will install itself as a service using systemd or init. Please remember to review the script itself to see if it fits your needs or system.

2 Video on Demand (VoD) 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/*.

2.1 Understanding Video on Demand (VoD) streaming

Vidoe on Demand streams have a set begging and end and and do not change over time. Unlike live streams that can add content indefinitely a VoD stream is set in stone, you will not be able to change or add to the content without essentially making a new/different VoD stream. As VoD files are usually created far before their planned release you have the time to polish your stream and make sure the content within is the best representation of what you wanted to achieve.

2.2 When do would I want to use VoD streaming?

If you have content that will not change or does not need to change VoD might just be the method for you. One of the biggest benefits to VoD streams is the ability to polish the stream as much as you want before you release it. You don't have to worry about getting it right ine one go, you can take multiple takes and combine them for the best possible content. While VoD streams are perfect for reaching a broad audience, as everyone can watch the content on their own preferred timeslot, it has one downside: Your content cannot grow. While not an issue for most streams, if you're not sure about the content and you want the possibility to change the content depending on events/circumstances you're probably better off with a live stream.

2.3 How do I create a VoD stream?

2.3.1 Generic instructions

Creating a VoD stream with MistServer is really easy, all you need to know is the location of the file you'd like to use as `source`. You'll need to create a new stream for each different `source`, if you'd like to add multiple `sources` and put them all available under different stream names at once you'll

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

probably be interested in the Pro only feature: folder support. You can create a stream in MistServer relatively easily, as long as your input and codecs are supported you'll have a stream up within a few minutes. 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.

2.3.2 Creating a stream in MistServer

To create a stream in MistServer you'll need to keep track of just two things: protocols and streams.

2.3.2.1 Setting up the protocols

Protocols are responsible for the way MistServer communicates with other devices, setting up the protocols can make or break your entire server. While easily adjusted, they can be difficult to custom edit. We'd advice to only change them if you're sure what you're doing, as for most users the standard options should suffice. If you've used the standard options and you do not want to custom edit the protocols you can skip this step.

To edit or add new protocols all you'll need to do is login to your MistServer at http://MistServerIP:4242 and go to protocols.

2.3.2.2 Setting up the stream(s)

You can edit the stream settings at the stream menu. The stream name is quite important as it's one of the `indentifiers` MistServer uses to place your stream at the right address. Simply browse to the file using the browse button or by typing the address in the bar. If you're running MistServer in a Cygwin environment (Windows), you'll need to use the following syntax to get to your files: *"/cygdrive/DRIVE_IN_CAPITAL/folder/file.extension"*an example would be the following: *"/cygdrive/C/testfiles/testmovie.flv"* this will get you to the file "testmovie.flv" in the folder "testfiles" in drive "C".

If you're uncertain about the codecs or compatability of your stream you could try encoding it to a different format. Please see our encoding to DTSC guide for the steps to re-encode towards DTSC format.

2.4 Folder support streaming

****MistServer Pro version only***

Folder support is our solution to adding multiple files at once and by having a dynamic availability system within MistServer. When you've chosen a folder to be used as a folder input MistServer will monitor that folder and add any new supported files as streams. Using the folder support option can save you a lot of time when setting up MistServer.

All you need to do to start using folder support is choosing a simple/short name as stream name and use the "select this folder" option in MistServer when setting up a stream or by filling in the folder as `source` and making sure the address ends on an *"/"*. Cygwin (Windows) users will need to use the Cygwin address: *"/cygdrive/DRIVE_IN_CAPITAL/folder/"*.

As soon as you've chosen the folder MistServer will automatically detect any supported files in that folder and put them available under the following address "stream_name+file.extension". Because the `stream name` is a vital part of your address we recommend using a short and easy to remember name.

As MistServer will still have to create a DTSH file for each stream and only creates one on the first stream attempt we recommend viewing each stream before making it available to other viewers. The time to create a DTSH file depends on the file size, so please wait up to 2 minutes for files ¿ 4GB.

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

3 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/*.

3.1 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.

3.2 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.

3.3 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.

3.3.1 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

3.3.2 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.

3.3.2.1 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.

3.3.2.2 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.

3.3.2.3 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.

3.3.2.4 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*

3.3.3 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.

3.3.3.1 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

3.3.4 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.

3.3.4.1 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

3.3.4.2 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.

3.3.4.3 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.

3.3.4.4 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

3.3.4.5 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*

3.3.4.6 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*

3.4 Multibitrate streaming

MistServer Pro version only

3.4.1 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.

3.4.2 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

3.4.3 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.

3.5 Wildcard streaming

****MistServer Pro version only***

3.5.1 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.

3.5.2 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.

3.5.3 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.

3.5.3.1 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

4 Embedding streams

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/*.

4.1 What is an embedded stream?

An embedded stream is simply putting a player in your website page that can play your chosen content.

4.2 Why would I want to embed streams?

Embedding a stream has one huge advantage: your viewers will not have to leave your website to view your content. You'll be able to keep your viewers on your own website and control the content they'll be able to view at all times. It's also easier for the viewers as they won't have to switch pages to view your content.

4.3 How do i embed a stream using MistServer?

Embedding with MistServer is quite easy, the protocol and preview menu contain all the information you'll need to embed your content.

4.3.1 Setting your stream not available message

You can change the stream not available message at the HTTP protocol in your protocols list. If you'd rather have a different message than the default you can change the `Stream not available text` in any message you'd like. We'd recommend to keep it short and understandable. You can use any `HTML` code you want in the message.

4.3.2 Using the embed information of the Preview menu

The embed information tab will help you embed your stream in a website. First make sure you set up all of the settings you'd want and then simply copy the embed code and paste it in your website. You've now embedded your stream. If you'd like to build your own embed code feel free to look at the embed info url as reference material. We're constantly looking for ways to improve the ease of use of our embed code so if you'd like to see another feature added to the embed tab feel free to contact us.

5 DTSC format

5.1 What is DTSC format

DTSC is a media format we've made for our media server. It's optimized for media distribution and can be instantly changed to fit outbound and inbound protocols.

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

5.2 Why or when should I use DTSC format

If you're using MistServer you're already using DTSC. MistServer converts any of the accepted inputs automatically to DTSC in shared memory/files. Your source won't be changed, but they will get a DTSH file with the same name in order for MistServer to not have to convert them a second time. Of course, you could speed up this process by changing a supported input to DTSC format which speeds up the streaming experience. The great thing about DTSC is that it's made to be flexible in both input/output and can support new protocols relatively easily.

5.3 How do I use DTSC files?

As soon as you use a supported input in MistServer you'll use DTSC automatically. MistServer will read your source and will add a DTSH file to easily convert it internally when used for streaming. You can speed this up by converting your file to DTSC format using our `MistIn` inputs.

5.3.1 Converting a file to DTSC

In order to convert a file to DTSC you will need to convert your file to a supported input/codecs. While the pro version has multiple inputs/codecs it supports, the open source version will only work with a few. So first you'll want to check whether your source is supported by MistServer without any encoding. If so you can go to step 2 immidiately.

5.3.1.1 Converting a file to a supported input

To create a source supported by MistServer you'll need to check both the input type and the codecs of your source. If you're not sure what codecs your file is in you can try converting it to a supported input type first, use it as a source file within MistServer and it will be able to tell you what codecs are used for your source file. Please review the supported codecs/input to decide the best fit for your streaming goal.
To convert a file to a supported input we recommend using ffmpeg. While somewhat hard to use at first due to the lack of an interface it's really efficient and fast to use once you've gotten used to it.
To convert a file towards a supported input open ffmpeg in a terminal and run the following command.

```
ffmpeg -i Input_File.ext -acodec copy -vcodec copy Output_File.output
```

To convert a file to a supported input type and codecs you can use the following command

```
ffmpeg -i Input_FILE.ext -acodec aac -ac 2 -strict experimental -vcodec libx264 \\
-profile:v baseline -level 30 Output_FILE.output
```

This command will create a new source file with AAC/H264 codecs, which gives the best cross protocol compatibility.

5.3.1.2 Converting a file to DTSC format

Now that we've created a supported input for MistServer we can either use this input file, or convert it to DTSC. Whilst instantly using would probably be more than enough for most use cases, those who want optimal speed for their VoD files can convert the files to DTSC. To do this you should use the corresponding `Mistin` with the input file. MistServer Pro has multiple `MistIns`, the open source version has just `MistInFLV`.

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

To use a `MistIn` you'll need to use a terminal and use a command line. Depending on your input and operating system your command line might change somewhat. Windows users will have to add `.exe` at the end of the `MistIn` for example. The following example is to convert a FLV file to DTSC under linux

```
MistInFLV Input_File.flv Output_File.dtsc
```

You will now have an `Output_File.dtsc` and an `Output_File.dtsh`, you will need both together as a valid DTSC file, make sure you choose the DTSC file as an input in MistServer.

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