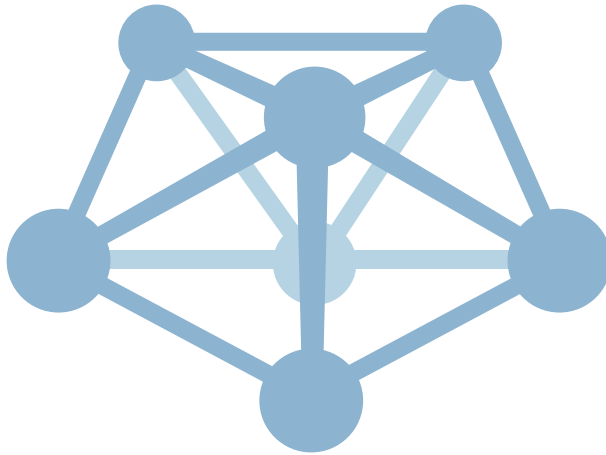


MistServer V2.11 Quickstart guide
DDVTech

May 17, 2017



MistServer V2.11 Quickstart guide



Contents

1 Quickstart guide for MistServer	3
1.1 List of files and their purpose	3
1.2 Running MistServer	3
1.2.1 Linux specific instructions	4
1.2.2 Windows specific instructions	4
1.2.3 MacOS specific instructions	4
1.3 Accessing the management interface and setting up MistServer	4
1.4 Enable protocols	4
1.5 Create streams	4
1.6 Viewing your streams	5
1.7 Standard stream url syntax	5



1 Quickstart guide for MistServer

1.1 List of files and their purpose

MistServer consists of multiple binaries, all working together to form the software as a whole. Each binary is dedicated to a specific task:

- **MistController**
This is the main binary, and the only mandatory one. It's functions include responding to API requests, discovering the other binaries, keeping track of statistics, logging, and starting and monitoring the various other binaries. It is the “brain” of MistServer, so to speak.
- **MistIn—**
These provide stream input capabilities, and their task is to act as the “source” of streams. Some of them read from files on the filesystem, others accept data through standard input or read data from a specific protocol connection. The specifics of each input are covered elsewhere in this manual.
- **MistInBuffer**
This is a special input: it expects live data to come in from some other location and maintains a buffer of the data, with metadata that is kept up-to-date. **Pro-only feature:** *MistInBuffer also performs “mixing” functions, allowing you to combine multiple separate live sources into a single stream.*
- **MistOut—**
These provide stream output capabilities, and their task is to communicate with the outside world. This can mean maintaining an open listening socket, or automated activation through another MistServer component. Additionally, some outputs are capable of writing to files and standard output.
- **MistAnalyser—**
These are not part of MistServer itself, and are not needed for the software to operate. They are utilities meant to assist in development and/or debugging and can provide information about the contents of buffers, various outputs, etcetera. Most users will not need to touch these under normal conditions.

It is safe to remove binaries that you do not plan on using, which will simply disable their related functionality in MistServer. Similarly, it is possible to write your own outputs and inputs to supplement MistServer's capabilities. For more information about this, please contact one of our engineers as it is outside of the scope of this document.

Besides the binaries, there is one more file: the configuration file. This file is a JSON-formatted plain text file, containing the complete setup of your MistServer installation. This file is loaded when MistServer starts, and written as MistServer exits. You can also force a manual write of the configuration through the API or the configuration interface. Its location can be controlled through a command line parameter, but its default location is `config.json` in the current working directory of MistServer. When running MistServer as a system service, we recommend using the location `/etc/mistserver.conf` instead.

1.2 Running MistServer

For the quickstart guide we will only discuss running without an installation. When MistServer is used for production we recommend installing MistServer, please use the full manual for installation help.



MistServer can be started without installing it, simply by extracting all its binaries to a single folder and executing the MistController. It will by default store its configuration in the current working directory.

It will walk you through a first-time setup on the command line, and then start listening for requests as well as make available the API port and configuration interface. Many choose to run MistServer in a screen session during testing or if they do not have root access to the machine MistServer needs to run on.

Do note that some of MistServer's automatic error recovery features will not function properly when not running as a system service.

1.2.1 Linux specific instructions

We recommend booting MistServer by the use of a terminal. Open a terminal window, browse to the folder where you've unpacked the binaries and run the MistController.

1.2.2 Windows specific instructions

For the best experience when using MistServer in Windows we recommend using the command prompt and running MistController.exe. If you are not familiar with the command prompt simply double click on MistController.exe. Several windows popping up is normal behaviour for MistServer in some versions of Windows, so do not get alarmed if this happens.

1.2.3 MacOS specific instructions

MistServer in MacOS requires permissions to create temporarily files, as such running MistServer as root is recommended. The easiest way to do this is to boot MistController from the Terminal using sudo. You can do this by opening the Terminal, typing sudo and dragging the MistController inside the terminal window. Make sure there is a space between sudo and the file path. You will be prompted to enter the root password before you can continue.

1.3 Accessing the management interface and setting up MistServer

Using the default settings Mistserver can be reached at `http://host:4242`. If ran locally you can always use `http://localhost:4242`. Once logged in you can browse the panels with the menu to the left, for the quickstart we will only discuss the protocols and streams panel.

1.4 Enable protocols

The protocols that are active will decide which protocols MistServer is able to use as either input or output. The default settings can be used by selecting *enable default protocols* should work for most users.

1.5 Create streams

Streams can be created by selecting *new stream* and require 2 key settings, *stream name* and *source*.

Stream name is used to identify the stream later on, so you know you will connect to the right stream when your users give this name to MistServer when connecting.

Source describes the input to be used, please view the help window that appears when you select the box for more information on the possible inputs.



1.6 Viewing your streams

Streams can be previewed from the management interface. In order to preview a stream go to the streams panel and press the *preview* button for the corresponding stream, you will be able to select multiple players and protocols.

For the direct stream urls and embeddable code for webpages please click the *embed* button.

1.7 Standard stream url syntax

Below is a table providing the standard url syntaxes for your convenience.

Protocol	Syntax
MP4	http://HOST:HTTP_PORT/STREAM_NAME.mp4
HLS	http://HOST:HTTP_PORT/hls/STREAM_NAME/index.m3u8
DASH	http://HOST:HTTP_PORT/dash/STREAM_NAME/index.mpd
RTMP	rtmp://HOST:RTMP_PORT/play/STREAM_NAME
	rtmp://HOST:RTMP_PORT/PASSWORD/STREAM_NAME
TS	http://HOST:HTTP_PORT/STREAM_NAME.ts
RTSP	rtsp://HOST:RTSP_PORT/STREAM_NAME
	rtsp://HOST:RTSP_PORT/STREAM_NAME?pass=password
HDS	http://HOST:HTTP_PORT/dynamic/STREAM_NAME/manifest.f4m
FLV	http://HOST:HTTP_PORT/STREAM_NAME.flv
HSS (Silverlight)	http://HOST:HTTP_PORT/smooth/STREAM_NAME.ism/Manifest
OGG	http://HOST:HTTP_PORT/STREAM_NAME.ogg