

# VisionLabs FaceStream

## Release notes

## Contents

<b>FaceStream v.5.1.18</b>	<b>4</b>
FaceStream developer's changes (v.5.1.17 - v.5.1.18) . . . . .	5
<b>FaceStream v.5.1.16</b>	<b>6</b>
FaceStream developer's changes (v.5.1.16) . . . . .	7
<b>FaceStream v.5.1.14</b>	<b>7</b>
<b>FaceStream v.5.1.13</b>	<b>8</b>
FaceStream developer's changes (v.5.1.13) . . . . .	9
LUNA Streams developer's changes (v.0.5.4 - v.0.5.5) . . . . .	9
<b>FaceStream v.5.1.12</b>	<b>9</b>
FaceStream developer's changes (v.5.1.11 - v.5.1.12) . . . . .	10
LUNA Streams developer's changes (v.0.5.3) . . . . .	10
<b>FaceStream v.5.1.10</b>	<b>10</b>
LUNA Streams developer's changes (v.0.5.1 - v.0.5.2) . . . . .	11
<b>FaceStream v.5.1.9</b>	<b>11</b>
LUNA Streams developer's changes (v.0.4.0 - v.0.5.0) . . . . .	12
<b>FaceStream v.5.1.8</b>	<b>13</b>
LUNA Streams developer's changes (v.0.3.0) . . . . .	13
<b>FaceStream v.5.1.7</b>	<b>14</b>
LUNA Streams developer's changes (v.0.2.2) . . . . .	14
<b>FaceStream v.5.1.6</b>	<b>15</b>
LUNA Streams developer's changes (v.0.2.0 - v.0.2.1) . . . . .	15
<b>FaceStream v.5.1.5</b>	<b>15</b>
<b>FaceStream v.5.1.4</b>	<b>16</b>
<b>FaceStream v.5.1.3</b>	<b>17</b>
<b>FaceStream v.5.1.2</b>	<b>18</b>
<b>FaceStream v.5.1.1</b>	<b>19</b>
<b>FaceStream v.5.1.0</b>	<b>19</b>

<b>FaceStream v.5.0.7</b>	<b>22</b>
<b>FaceStream v.5.0.6</b>	<b>22</b>

## FaceStream v.5.1.18

- Implemented functionality for collecting and sending FaceStream monitoring data to the Influx database. Monitoring for FaceStream is disabled by default. To enable FaceStream monitoring, you need to enable the “send\_data” setting and set additional settings from the “monitoring” section of the FaceStream settings.

The following data is sent to InfluxDB:

- measurement element. It is equal to the value of `fs-requests`.
- tag set:
  - \* `fs_ip` - IP address where FaceStream is deployed
  - \* `source` - the “name” field set when creating a stream in LUNA Streams (optional)
  - \* `stream_id` - stream ID
- field set:
  - \* `track_id` - track ID
  - \* `event_id` - event ID received from LUNA PLATFORM
  - \* `request_id` - external ID for communication with monitoring of LUNA PLATFORM services
  - \* `track_start_time` - track start time
  - \* `track_best_shot_time` - the time when the frame with the bestshot being sent appeared in the system
  - \* `track_best_shot_min_size_time` (optional) - the time when the detection size reached the value specified in the “best\_shot\_min\_size” parameter
  - \* `track_best_shot_proper_size_time` (optional) - time when the detection size reached the value specified in the “best\_shot\_proper\_size” parameter
  - \* `liveness_start_time` (optional) - Liveness start time
  - \* `liveness_end_time` (optional) - Liveness end time
  - \* `bestshot_count` - the number of bestshots sent
  - \* `time_from_first_frame_to_send` - the time that passed from the appearance of the first frame in FS to sending to LP
  - \* `track_send_count` - the number of tracks sent

Tags containing time are sent as UTC with microsecond precision.

- `<timestamp>` element. Is the time the bestshot(s) was(were) sent in microseconds.

See the “[Monitoring](#)” section of the administrator manual for more information.

- Added user interface for the LUNA Streams service.

The interface can be opened in a browser by specifying the address and port of the LUNA Streams service: `<streams_server_address>:<streams_server_port>`

The user interface of the service contains three tabs - “Streams”, “Groups” and “Queue”.

- “Streams” is a tab where streams statuses and their preview are displayed, it is possible to create and manage streams.
- “Groups” is a tab where streams groups are displayed.
- “Queue” is a tab where the streams that are in the processing queue are displayed.

See the detailed information in the [“LUNA Streams user interface”](#) section of the administrator manual.

- Implemented sending coordinates of bounding rectangles of faces (X, Y, width, height) when using an image without transformation (setting `portrait_type=gost`). Sending is implemented by adding the `face_bounding_boxes` parameter containing an array of bounding boxes corresponding to the detected faces.

The coordinates of the rectangle are defined in the portrait coordinate system.

The coordinates of the rectangle are saved in the `face_bounding_boxes` field of the event generation request body. These coordinates are used when generating an event, which allows face redetection, which is performed faster than face detection.

- The maximum size of batch with sets of images and detections sent for AGS and HeadPose estimation is limited to 16.

This reduces the memory consumption of the FaceStream.

#### **Fixed errors**

- Fixed the error where the value of `silent_period=-1` could cause the track to be sent several times.

This error caused multiple events with the same `track_id` to be generated in LUNA PLATFORM.

### **FaceStream developer's changes (v.5.1.17 - v.5.1.18)**

#### **Improvements**

- Added facility for collecting and sending monitoring data.
- The maximum batch size that is transferred to estimators is limited to 16.
- Add sending the `face_bounding_boxes` parameter in generate event request to Luna. The parameter contents the array of elements corresponded to detected faces in GOST portrait mode. Each element is consisted from image file name and the rectangle { x, y, width, height} bounded the detected face. The coordinates of rectangle is defined in coordinate system of portrait.

## Fixed bugs

- Fixed a bug due to which the track could be sent multiple times with `silent_period = -1` parameter.

## FaceStream v.5.1.16

- Implemented sending coordinates of bounding rectangles of faces (X, Y, width, height) when using a sample as an image source (setting `portrait_type=warp`). Sending is implemented by adding the `face_bounding_boxes` parameter containing an array of bounding boxes corresponding to the detected faces.

The coordinates of the bounding box are defined in the coordinate system of the source image.

Support for saving coordinates to an event is implemented in LUNA PLATFORM v.5.36.5. Thus, starting from this version of the FaceStream build, it is required to use a LUNA PLATFORM version of at least v.5.36.5.

The coordinates of the bounding box are stored in the `“face_detections” > “detection” > “rect”` field of the LUNA PLATFORM event.

- Manuals for installing and updating FaceStream were divided into two parts:
  - manuals for installing and updating FaceStream with LUNA PLATFORM launched. It is assumed that LUNA PLATFORM was launched according to the documentation from LUNA PLATFORM distribution where default paths are used. These manuals are located in the `“/docs”` directory.
  - manuals for installing and updating FaceStream without LUNA PLATFORM launched (with license activation and additional launch of Licenses, Configurator services and PostgreSQL and Influx databases). These manuals are located in the `“/docs/FS_without_LP”` directory.

## Fixed errors

- The error was fixed where the `“callback-mode”` parameter of the TrackEngine settings was not read. FaceStream only works with `“callback-mode=0”`. Attempts to set the value of `“callback-mode=1”` will result in a warning message in the FaceStream logs and the value of `“callback-mode=0”` will be used.
- Reduced generation time of data in JSON format and implemented sequential processing of frames.

This resulted in a fix for a error where using the `“json”` value to set `“sending > request_type”` would spike memory usage.

**Note.** Long time to generate JSON data can lead to increased memory consumption.

- The error was fixed in which the mode of sending images of detected faces without transformation (the `“portrait_type=gost”` parameter) didn't work on the GPU.

## LUNA Streams fixed errors

- The error was fixed where “location” > “geo\_position” > “latitude”/“longitude” fields could not be set to integers.

Now you can specify both integer and float numbers.

## FaceStream developer's changes (v.5.1.16)

### Improvements

- Add sending the face\_bounding\_boxes parameter in generate event request to Luna. The parameter contents the array of elements corresponded to detected faces in warp portrait mode. Each element is consisted from image file name and the rectangle { x, y, width, height} bounded the detected face. The coordinates of rectangle is defined in source image coordinate system.

### Fixed bugs

- Fixed a problem when parameter callback-mode was not read from configurator. FaceStream read settings from file and configurator. FaceStream always works with callback-mode = 0. If the value 1 is set then the warning message appears and value 0 is still used.
- Reduced json generation time for sending/request\_type: “json” and made frame processing sequential. Long json generation time may lead to enormous memory consumption.

## FaceStream v.5.1.14

- FSDK was updated to version 5.12.0.

In this version, the Liveness Flying Faces estimator has been updated.

- The description of many parameters in the [LUNA Streams specification](#) has been expanded.

### Fixed errors

- The error has been fixed where FaceStream could send an incorrect preview URL in the “preview > live > url” field of the LUNA Streams stream, which caused the following error and FaceStream continued to work:

```
Failed to validate input json' to occur in the FaceStream logs. Path: '
streams.0.preview.live.url', message: 'URL host invalid',
description: Bad/incomplete input data.
```

This error could occur due to the fact that LUNA Streams was started after FaceStream (for example, the server was restarted, where both FaceStream and LUNA Streams are running), and in order to determine its IP address, the FaceStream application needs to either connect to the LUNA Streams service, or read the predefined IP address in the system environment variable “VL\_FACE\_HOST” (see section “[Nuances of working with stream preview](#)” of the administrator manual).

Now if the IP address of FaceStream is determined by connecting to LUNA Streams and the service starts after FaceStream, then FaceStream will terminate with the error “Failed to get local IP address. Reason ...”.

If the IP address of FaceStream is determined using the correctly set “VL\_FACE\_HOST” environment variable, then such an error will not occur.

- The error has been fixed where the DROI bounding box flashed while viewing a stream in the browser if “[show\\_bbox](#)” was set to “0”.
- The error has been fixed where some detections with human body coordinates for the best shots were not sent to LUNA PLATFORM.

Accordingly, these detections were absent in events generated using the dynamic handler specified in the “detection\_handler” parameter.

## FaceStream v.5.1.13

- A mechanism for migrating LUNA Streams settings set in the LUNA Configurator service has been added.

### LUNA Streams update

Previously, LUNA Streams settings were automatically migrated when migrating the LUNA Configurator service. Now LUNA Streams settings are migrated using a separate migration mechanism. See the upgrade manual for more details.

### Launch LUNA Streams from scratch

Now the LUNA Streams settings are loaded into the Configurator service using the migrations mechanism. The FaceStream settings are still loaded into the LUNA Configurator using the “load\_dump.py” script. See installation manual for more details.

### Fixed errors

- FaceStream crash has been fixed that occurred when an incorrect address was specified in the “lunastreams” > “origin” parameter of the FaceStream settings.

Now the FaceStream operation is finished correctly, and the corresponding error is written to the logs.



- The error has been fixed where the valid range (from 0 to 1024 pixels) was not checked for the “size\_source\_frame” parameter of the FaceStream settings, resulting in the possibility of setting any values for this parameter.
- The error has been fixed where FaceStream would output a stack trace instead of an error if the port it was running on was busy.

Now the error `Error occurred on initializing application. Description: bind: Address already in use` is displayed in the FaceStream logs.

- The error has been fixed where changing the “enable\_gpu\_processing” setting of the FaceStream settings was not applied properly and caused a crash when the configuration was reloaded.

## FaceStream developer’s changes (v.5.1.13)

### Fixed bugs

- Fixed HTTP server crash on used port.
- Fixed a bug where changing the “enable\_gpu\_processing” option did not applied properly and lead to crash on config reloading.
- Fixed a bug where invalid LunaStreams URL in lunastreams/origin lead to crash.
- Fixed a bug on checking the resize source frame parameter in config. Add the checking of allowed range of parameter.

## LUNA Streams developer’s changes (v.0.5.4 - v.0.5.5)

### Improvements

- The settings migration mechanism was updated.
- Service dependencies were updated.

## FaceStream v.5.1.12

- FSDK was updated to version 5.11.0.

### Fixed errors

- The error was fixed that caused the stream error counter to be processed incorrectly. If during the “period” the number of errors did not reach the value “max\_error\_count”, then the error counter was not reset, but continued to increase.
- The error was fixed, due to which zero-length strings and “null” values from the “location” section of streams management settings were written to the LUNA PLATFORM event.

## FaceStream developer's changes (v.5.1.11 - v.5.1.12)

### Improvements

- FSDK to v.5.11.0

### Fixed bugs

- Fixed a bug where stream's error counter was handled incorrectly.
- Fixed a bug where empty strings and null fields of "location" was treated equally.

## LUNA Streams developer's changes (v.0.5.3)

### Improvements

- Service dependencies were updated.

## FaceStream v.5.1.10

- Support for the new LUNA PLATFORM 5 authorization system was added.

New optional authorization field "event\_handler > authorization" was supported in the [stream creation settings](#), in which you need to specify "token" or "account\_id". The "event\_handler" > "authorization" > "account\_id" parameter must match the "account\_id" parameter set when creating the stream. If the new authorization field is not filled in, the "account\_id" set when creating the stream will be used.

When upgrading to new version of LUNA Streams, the database will be automatically migrated to support the new field.

**Starting from this version of the FaceStream build, it is required to use the LUNA PLATFORM version at least v.5.32.0.**

- LUNA Streams and FaceStream service containers are now named the same when manually launched and when launched with Docker Compose.
- The Python version was updated to 3.10 in the LUNA Streams and LUNA Configurator containers.  
All commands related to using Python inside containers were updated in the documentation, namely, commands like "python3.9" were replaced with "python3".
- New documents "FS\_Docker\_Upgrade\_Manual\_Eng.pdf" and "FS\_Docker\_Upgrade\_Manual\_Eng.html" were added to the distribution package, describing the upgrade process from the previous build of LUNA Streams and FaceStream services, as well as LUNA PLATFORM services launched in accordance with the FaceStream documentation.

## Fixed errors

- The error was fixed where the “jpeg\_quality\_level” setting was not working correctly on the GPU. With the “request\_type” = “jpeg” value, only the first source frame was compressed. The rest of the source frames were not compressed.  
With the “request\_type” = “json” value, the source frames were not compressed, but all warps were compressed, except for the first one.

## LUNA Streams developer's changes (v.0.5.1 - v.0.5.2)

### Improvements

- Service dependencies were updated.
- Stream description length has been increased from 128 to 512 symbols, [create stream](#) for details.
- Support for older python versions was dropped. Python 3.10 is required now.

## FaceStream v.5.1.9

### FaceStream changes

- DROI calculation mechanism was changed.  
Previously, DROI represented coordinates relative to ROI, taking into account image rotation. DROI now represents coordinates relative to the original frame.
- The FaceStream stream processing report regularly sent to LUNA Streams was expanded with new “last\_frame” and “live” fields containing the URLs of the image's last frame and the live stream.  
These URLs can be received using specific requests to the LUNA Streams service (see below).
- FaceStream operation was accelerated and GPU memory consumption was reduced by limiting the estimation of five landmarks in cases where such estimation is not required.
- Displaying the IP address of the server running FaceStream was added to the FaceStream logs.

### LUNA Streams service changes

- Migration was added to LUNA Streams, in which all existing streams will be updated in accordance with the new DROI calculation mechanism (see FaceStream changes above).  
When updating FaceStream, it is necessary to update LUNA Streams.
- A “include\_luna\_services” parameter was added to [“get health”](#) request.  
The “include\_luna\_services” parameter allows you to enable or disable healthcheck for the LUNA PLATFORM services that the LUNA Streams service depends on. If this option is enabled, then additional requests are sent to the “/healthcheck” resources of these services.

- Two new resources were added - [“get last frame preview”](#) and [“get live preview”](#) that allow you to get the values of the “last\_frame” and “live” fields from the FaceStream stream processing report (see FaceStream changes above).

New parameters “preview” > “last\_frame” and “preview” > “live” were added to the [“get streams logs”](#) request.

- The automatic restart of the stream now has its own statuses (the “status” parameter).

The automatic restart statuses are listed below:

- “disabled” - stream automatic restart is disabled by user (“restart” parameter is disabled)
- “enabled” - automatic restart is enabled but is not currently active because the stream is not in the “failure” status
- “in\_progress” - automatic restart in progress
- “failed” - the allowed number of automatic restart attempts was exceeded and none of the attempts were successful
- “denied” - automatic restart allowed by user, but not possible due to fatal error received in the FaceStream report. At the moment, the critical error has the following content: “Failed to authorize in Luna Platform”. In future versions, the list of critical errors may be updated.

The status of the automatic restart can be obtained using the [“get stream”](#) request.

## LUNA Streams developer’s changes (v.0.4.0 - v.0.5.0)

### Improvements

- The DROI calculation mechanism was changed.

Previously, DROI represented coordinates relative to the ROI, taking into account the rotation of the image.

From now on, DROI represents coordinates relative to the full frame.

All existing streams will be updated according to the new DROI calculation mechanism.

- The `live` and `last_frame` stream preview were added.

See [last frame preview](#) and [last frame preview](#) for details.

- The stream *autorestart status* was added. The stream *autorestart* mechanism was updated - the fatal error concept was added.

For more information see *autorestart* section of [get stream](#) response and streams automatic restart.

- The `include_luna_services` parameter was added for [healthcheck](#) resource.
- New field `preview` for stream feedback was added. See [stream feedback](#) for details.

## FaceStream v.5.1.8

- The script for uploading FaceStream and LUNA Streams dumps to the LUNA Configurator service can now be run on Python 2.x and Python 3.x versions. To run the script on the Python 2.x version, you must use the “python” command. To run the script on Python 3.x, you must use the “python3” command. Also, the script for loading dumps was added to Docker Compose. Now, to use the “start\_facestream.sh” script, you only need to have running LUNA PLATFORM containers and an activated license.
- Support for Liveness on the GPU was added.

### FaceStream fixed errors

- The error was fixed where the “Streams total” counter in the FaceStream logs counted streams with a “failure” status.
- The error was fixed that was causing FaceStream crashes in a multi-threaded environment.
- The error was fixed where image rotation, controlled by the “rotation” setting, was performed clockwise on the CPU and counterclockwise on the GPU. Rotation is now clockwise on both types of processors.
- The error was fixed, in which FaceStream took more streams into processing than specified in the license key.
- The FaceStream crash was fixed that occurred when setting a non-existent tag in the “logging” > “tags” field in the FaceStream settings.
- The FaceStream crash was fixed that occurred when running on a GPU with “size\_source\_frame” from FaceStream settings set to non-zero.

### LUNA Streams service changes

- Significantly accelerated performing of requests to “[get stream](#)” and “[get streams](#)” resources with a large number of entries in the stream log tables of the LUNA Streams database.
- New resource “[delete streams logs](#)” was added, which enables you to delete the streams logs older than the date specified in the request (the “log\_time\_\_lt” parameter), excluding the last entry in the log.

## LUNA Streams developer’s changes (v.0.3.0)

### Improvements

- The [getStream](#) and [getStreams](#) requests have been accelerated in the presence of a large number of logs.
- Added possibility to delete streams logs. See [delete streams logs](#) for more info.

- New field `authorization` for stream event handler was added.

## FaceStream v.5.1.7

- Now the URL for sending the source frames of faces and bodies is specified in the “`frame_store`” parameter. Previously, the URL from the “`origin`” parameter was used to send the source frames of the bodies, and the URL to send the source frames of the faces was set in the “`frame_store`” parameter.
- Now, for failed requests to LUNA PLATFORM services, the FaceStream logs will show the “`Luna-Request-Id`” value generated by FaceStream.

Based on the generated “`Luna-Request-Id`” value, you can find an error in the logs of LUNA PLATFORM 5 services.

- A script “`load_dump.py`” was added to the “`example-docker/luna_configurator/dumps/`” directory of the distribution package, which enables you to upload FaceStream and LUNA Streams settings to LUNA Configurator.

Example of command to upload settings for LUNA Streams service: `python -m load_dump --dump-file=streams_dump.json --luna-config=http://127.0.0.1:5070/1`

This method is taken as the main one in the FaceStream installation manual (see the “Uploading settings to LUNA Configurator” section).

### FaceStream fixed errors

- The crash was fixed that occurred due to incorrect handling of server errors when sending the source frame.
- The error was fixed, which led to a significant increase in RAM consumption during FaceStream operation.

### LUNA Streams service changes

- New parameters “`duration`” and “`progress`” were added to the “`video_info`” field of the response to a request to get information about stream (“[get stream](#)” request).

The “`duration`” parameter displays the total duration of the videofile in seconds, and the “`progress`” parameter displays the current videofile progress from 0 to 1, where 0 is the beginning of the videofile, 1 is the end of the videofile.

## LUNA Streams developer’s changes (v.0.2.2)

### Improvements

- Service dependencies were updated.

## FaceStream v.5.1.6

- The “send\_detection\_path” parameter was removed from the FaceStream settings. Now the ability to send detections with the coordinates of the human body is controlled by the numerical value of the “minimal\_body\_track\_length\_to\_send” parameter. If the “minimal\_body\_track\_length\_to\_send” value is “0”, then no detections will be sent.
- Now, when sending a source frame to the LUNA Image Store using the “frame\_store” parameter, the account ID will also be sent. The <source\_frame\_id>.meta.json file containing the “account\_id” parameter will be saved in the folder with the source frame in the bucket.

### LUNA Streams service changes

- The error was fixed, in which after transferring the stream to the “pause” status, the status “in-progress” was displayed in the stream logs (see the [“get streams logs”](#) request).
- The error was fixed that allowed setting very large values for parameters in [“create stream”](#) and [“Post feedback on streams”](#) requests.

## LUNA Streams developer’s changes (v.0.2.0 - v.0.2.1)

### Improvements

- Service dependencies were updated.
- New fields duration and progress for stream feedback was added.

### Fixed bugs

- Fixed bug which allowed to set very large values to stream data, stream policies and stream feedback.

See [createStream](#) and [postStreamsFeedback](#) documentation for more info.

- Fixed bug which caused extra entry to appear in stream logs after pausing the stream. See [documentation](#) for more info.

## FaceStream v.5.1.5

- The ability to get the detection time (in seconds) of a face or body relative to the start of the video was added.

This time is stored in the “detect\_ts” field of the LUNA PLATFORM event.

Getting face or body detection time is available only for the “videofile” source type.

- LUNA Streams version was updated.

### FaceStream fixed errors

- The error was fixed where the stream would not go into “failure” status when invalid parameters were used.
- Getting the “detections\_count” metric on the GET [“/metrics”](#) request was fixed.
- The “show\_window” parameter, deprecated since version 5.0.6, was removed from the FaceStream settings.
- The error was fixed that occurs on some graphics cards when previewing a stream using the [“/streams/preview/{id}”](#) resource while using FaceStream with GPU.

### LUNA Streams service changes

- New “endless” parameter was added to the [“create stream”](#) and [“put stream”](#) requests of the LUNA Streams service, which enables you to control the restart of the stream when a network error is received (the error is determined by the system as an eof (end-of-file) marker).

The parameter is available only for the “udp” and “tcp” source types.

If the `endless` parameter takes the value `true`, then in case of receiving eof and successful reconnection, the processing of the stream will continue. If all reconnection attempts failed (see the [“healthcheck”](#) section), then the stream will take the “failure” status. If the parameter takes the value `false`, then the processing of the stream will not continue and the status of the stream will take the “done” status.

When using a video file as a “tcp” or “udp” source, it is assumed to use the value `false`. This will avoid re-processing an already processed fragment of the video file when receiving eof. If, when using a video file, the value of the `endless` parameter is `true`, then after the processing is completed, the video file will be processed from the beginning.

### LUNA Streams service fixed errors

- The order of getting stream logs in the [“get streams logs”](#) request was fixed. Now the new logs are shown first.

## FaceStream v.5.1.4

- Two new launching keys for FaceStream were added - “CONFIG\_RELOAD” and “PULLING\_TIME”.

The “CONFIG\_RELOAD” launching key enables checking for changes in the “FACE\_STREAM\_CONFIG” section of the LUNA Configurator service and takes the following values:

- “1” - change tracking is enabled, if there are changes in the configuration, all FaceStream containers will be automatically restarted;



- “0” - change tracking is disabled (default value).

The “PULLING\_TIME” launching key sets the period for receiving new parameters from the “FACE\_STREAM\_CONFIG” section of the LUNA Configurator service in the range [1...3600] sec. Used in conjunction with the “CONFIG-RELOAD” tag. The default value is “10”.

Launching keys are also available in FaceStream mode with configuration files.

- New parameter “min\_body\_size\_threshold” was added to the “filtering” section in the streams management settings.

The parameter sets the body detection size, less than which it will not be sent for processing. It is calculated as the square root of the product of the body detection height (in pixels) by its width (in pixels).

Example:  $\text{min\_body\_size\_threshold} = \sqrt{64 \times 128} = 90.5$

If the value is “0”, then filtering of body detection by size will not be performed.

### Fixed errors

- The address of the source image is now recorded in the “detections” > “image\_origin” field of the LUNA PLATFORM event. Previously, this address was recorded in the “face” > “user\_data” field.
- The error was fixed, in which the “limit” parameter (maximum number of the streams to return) was not processed in the request to get streams from the queue (“/1/streams/processing/queue” resource) when filtering by group.

## FaceStream v.5.1.3

- SDK was updated to version 5.6.0.
- The ability to group streams was added. Grouping is intended to combine streams with multiple cameras into logical groups. For example, you can group streams by territorial characteristic.

Any stream can belong to several groups, as well as none.

The group is created using the “[create group](#)” request. To create a group, you need to specify the required parameters “account\_id” and “group\_name”. If necessary, you can specify a description of the group.

Stream can be linked to a group in two ways:

- using the “group\_name” or “group\_id” parameters during stream creation (“[create stream](#)” request).
- using the “[linker](#)” request. In the request, you should specify the streams IDs and the group to which they need to be linked. You can also unlink streams from a group using this request.

If the stream was linked to a group, then the “get stream” or “get streams” requests will show the group in the “groups” field.

- The ability to transfer local track timestamps to events created by LUNA PLATFORM 5 was added.
  - For an event created by a static handler, the “create\_time” and “end\_time” parameters are now recorded.
  - For an event created by a dynamic handler, the “end\_time” parameter is now recorded.
  - For the bestshots (samples), the “detect\_time” parameter is now recorded.
- Now, when the “send\_detection\_path” parameter is enabled, the corresponding detection with body coordinates will be sent along with the bestshot by default. Previously, such a detection could not be sent.

## FaceStream v.5.1.2

- The parameter “preferred\_program\_stream\_frame\_width” parameter was added to the streams management settings, intended to work with protocols that imply the presence of several channels with different bitrates and resolutions (for example, HLS).

If the stream has several such channels, then this parameter will enable you to select from all the channels of the whole stream the channel whose frame width is closer to the value specified in this parameter.

For example, there are 4 channels whose frame widths are 100, 500, 1000 and 1400. If the parameter “preferred\_program\_stream\_frame\_width” is equal to “800”, then a channel with a frame width of 1000 will be selected.

If the stream has only one channel, this parameter will be ignored.

This parameter is used only for “tcp” or “udp” types.

- The following new parameters were added to the “lunastreams” section of FaceStream settings:
  - request\_stream\_period - parameter sets the time period between requests to receive new streams from LUNA Streams in the range from 0.1 to 3600 seconds.
  - send\_feedback\_period - parameter sets the time period between sending reports on processed streams to LUNA Streams in the range from 1.0 to 3600 seconds. The value of this parameter should not exceed the value of the “STREAM\_STATUS\_OBSOLETING\_PERIOD” parameter, set in the LUNA Streams service settings.
  - max\_feedback\_delay - parameter sets the maximum report sending delay in the range from 1.0 to 3600 seconds. If the report has not been sent within the given time, then FaceStream will stop processing the current streams. The value of this parameter should not be less

than the value of the parameter “send\_feedback\_period” and should not exceed the value of the parameter “STREAM\_STATUS\_OBSOLETING\_PERIOD”, set in the LUNA Streams service settings.

## FaceStream v.5.1.1

- The error in the `docker-compose.yml` file that caused the error “host network\_mode is incompatible with port\_bindings” when running a Docker Compose script was fixed.

## FaceStream v.5.1.0

- The principle of creating streams was changed. Previously, settings for streams were specified in the “input.json” configuration file or in the “FS\_CONFIG” section of the Configurator. Now the settings for streams are set in the body of the JSON format and sent using an HTTP request to the LUNA Streams service (see the “[create stream](#)” request in the “StreamsReferenceManual.html” document of the distribution).

After sending an HTTP request with the given parameters to the LUNA Streams service, the parameters are added to the LUNA Streams database under the unique identifier “stream\_id”. Depending on the number of available streams in the license (this information comes from the LUNA Licenses service), the stream parameters enters the internal queue, where it is in the “pending” status until FaceStream takes them from the queue for further processing (see the detailed description of the service in the “Interaction of FaceStream with LUNA Streams” in the administrator manual).

### Licensing changes:

The service can process a certain number of streams simultaneously. The number of such streams is set in the LUNA PLATFORM 5 license key parameter, which determines the streams number for LUNA Streams operation. Thus, starting from version 5.1.0, FaceStream is licensed using the LUNA PLATFORM 5 key.

To launch LUNA Streams and FaceStream, you need at least three additional LUNA PLATFORM components:

- LUNA Licenses service. This service is used to license the LUNA Streams service.
- LUNA Configurator service. LUNA Streams settings are stored in this service in the same way as “FACE\_STREAM\_CONFIG” and “TRACK\_ENGINE\_CONFIG” settings for FaceStream. If you do not need to use LUNA Configurator, you can launch LUNA Streams using configuration files, but these steps are not described in the installation manual.
- PostgreSQL container. In this container, a LUNA Streams database is created, in which the stream parameters specified earlier in the JSON body are stored (see the description of the

database in the “LUNA Streams database description” section of the administrator manual). Optionally, you can use an Oracle database. Its use is not described in the documentation.

The InfluxDB OSS 2 container is also launched by default. Using the Influx database, LUNA PLATFORM services are monitored. If necessary, monitoring can be disabled.

If the LUNA PLATFORM components are not launched and the LUNA PLATFORM license is not activated, then the following steps should be performed before launching FaceStream:

- License should be activated.
- Required components should be launched.
- LUNA Streams and FaceStream settings should be uploaded to LUNA Configurator.
- LUNA Streams database should be created and initialized.

If the LUNA PLATFORM components are launched and the LUNA PLATFORM license is activated, then the following steps should be performed before launching:

- LUNA Streams and FaceStream settings should be uploaded to LUNA Configurator.
- You should make sure that the license contains a parameter that determines the number of streams to be processed by the LUNA Streams service and specify the address of the server launched the LUNA Licenses service in the LUNA Configurator service (if the LUNA Licenses service is not on the server launched LUNA Streams).
- LUNA Streams database should be created and initialized.

For a description of how to perform these steps, see the FaceStream installation manual.

- The ability to deploy FaceStream using a Docker Compose script was added. The script enables you to automatically launch and configure LUNA Streams and FaceStream. Before running the script, the following steps should be performed:
  - License should be activated.
  - Required components should be launched.
  - LUNA Streams and FaceStream settings should be uploaded to LUNA Configurator.
  - LUNA Streams database should be created and initialized.
- Description of errors returned by the LUNA Streams and LUNA Licenses services, as well as general errors that may occur when interacting with the LUNA PLATFORM, was to the documentation.
- The “lunastreams” section was added to the FaceStream settings (“FACE\_STREAM\_CONFIG” section of the LUNA Configurator service), enabling you to specify the address of the LUNA Streams service used (“origin” parameter), its API version (“api\_version” parameter) and the upper bound on the number of FS streams (“max\_number\_streams”). If LUNA Streams service is launched on another server, then you should specify the current address of the service before launching FaceStream. See the FaceStream administrator manual for details.
- The names and structure of streams management settings was reworked. All settings are now divided into logical blocks. The following settings was renamed:

- input, luna\_account\_id > account\_id
- input, transport > data, type
- input, url > data, reference
- output, image\_store\_url > event\_handler, frame\_store

The method of passing data to generate an event was reworked. Previously, to generate a LUNA PLATFORM event, the “output” > “url” parameter was used, in which, when working with faces, it was necessary to specify the address to the “/events” resource in the format `http://<luna_address>:<port>/6/handlers/<handler_id>/events`, where <handler\_id> is the LP handler identifier. When working with bodies, separate parameters “luna\_human\_handler\_id” and “luna\_dynamic\_human\_handler\_id” were used, and the “url” field was filled in the format `http://<luna_address>:<port>/6/`. Now the handler ID is set in the “event\_handler” > “bestshot\_handler”/“detection\_handler” > “handler\_id” group, where “bestshot\_handler” is a static handler for a face or body, “detection\_handler” is a dynamic “handler\_id” LUNA PLATFORM for working with bodies (ex. “luna\_dynamic\_human\_handler\_id”). The LUNA API service address and API version are now set separately. Below is an example of passing data to generate an event when static and dynamic handlers are set for the body.

```
"event_handler": {
  "origin": "http://127.0.0.1:5000",
  "api_version": 6,
  "bestshot_handler": {
    "handler_id": "ee4c42b6-23ae-410e-a2aa-a4220e64ba4b"
  },
  "detection_handler": {
    "handler_id": "426542d6-5509-4e5b-8a01-e2abd5c0a8c6"
  }
}
```

If it is necessary to process a face, then “handler\_id” should be set with parameters for processing a face in the “bestshot\_handler” group, and the “detection\_handler” group is optional.

See the FaceStream administrator manual for details on stream management settings.

- Support for normal FaceStream launch mode was dropped. Now you can launch FaceStream only in server mode.
- Information on launching FaceStream in server mode with configuration files was moved to the administrator manual (see “Use FaceStream with configuration files” section).
- The default logging parameters was changed in the FaceStream settings, in the “Logging” section:
  - The “severity” parameter is now equal to “1”. This means that only system warnings will be displayed in the logs.

- The “mode” parameter is now equal to “l2c”. This means that logs will be output only to the console.

## FaceStream v.5.0.7

- SDK was updated to version 5.5.0.
- The ability to get the current version of FaceStream using the GET “/version” request was added (see the detailed information in the “FaceStreamApi.html” document).
- Fixed an error where when running FS in server mode with Configurator, the “save-debug-info” parameter from the trackengine configuration was not used.
- Fixed an error where the source frame without detections with human body coordinates (detection\_path) was not sent with the “send\_source\_frame” parameter set.
- Fixed an error where sending “track\_id” to faces did not work, which is why the value of “track\_id” was not recorded in the LUNA PLATFORM event.
- Fixed an error when using the mouth estimator. The error could occur when using two streams, when the first one used this estimator, and the second one did not.

## FaceStream v.5.0.6

- FaceStream 5 is now distributed in Docker containers only. CentOS 8 is used for running FaceStream inside the container.
- FaceStream distribution package was updated. Now it includes documentation and configuration files required for the FaceStream launch.
- The “LP\_Docker\_Installation\_Manual\_Eng” document was added. It describes FaceStream launching in Docker container. The information is moved from the administrator manual. The document is provided in PDF and HTML formats.
- SDK was updated to version 5.4.1.
- Support for working with bodies was added. FaceStream searches for bodies in the stream and tracks them until they leave the frame or are overlapped.

There are additional settings for using FaceStream with bodies. Starting with FaceStream 5.0.6, all settings are divided into common and individual settings for faces and bodies in the documentation.

To enable body tracking, you need to change the “use-face-detector” parameter of the “trackengine.conf” configuration file to “use-body-detector” and also set individual settings for stream sources for bodies (see the relevant chapters).

The logic of creating events (sending bestshots and processing them by LUNA PLATFORM) for the body differs from the logic of creating an event for faces (see the administrator manual for details).

- All parameter names in configuration files were unified. For example, the names “frame-processing-mode” and “real\_time\_mode\_fps” were used. Now all the names look the same: “frame\_processing\_mode” and “real\_time\_mode\_fps”.
- Error messages were updated. Now they are more detailed and contain the path to the erroneous parameter.
- Synchronous waiting for a response from LUNA PLATFORM (also affects the performance of the working with bodies) was removed.
- The server time will now be sent with the bestshot. Previously, the device time was sent. This feature applies to all types of sources (video streams, video files and sets of images).
- Support for LUNA PLATFORM API versions 4 and 5 was discontinued. FaceStream cannot send data to LUNA PLATFORM 3, LUNA PLATFORM 4, Backport 3, and Backport 4 starting with this version. Sending images is only possible in LUNA PLATFORM 5 (API version 6).
- Using the “show\_window” parameter and its associated parameters is no longer possible.