

VisionLabs FaceStream

Release notes

Contents

FaceStream v.5.1.2	3
FaceStream v.5.1.1	3
FaceStream v.5.1.0	3
FaceStream v.5.0.7	6
FaceStream v.5.0.6	7

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 was 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_adress>:<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_adress>:<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.