## Latest Release

**Version: v0.0.20**

**Features/changes:**

* Added data storage management feature, which will delete uploaded blocks, if memory usage is more than 90 and also, it will show alerts to users.
* Added more streaming parameters configurable through CLI, they are,
  + Buffer duration in seconds
  + Block duration in seconds
  + Chunk duration in seconds
* Check the help in CLI for more information on new parameters
* Added Amplification feature in oedge.
  + Users can see the current amplification value on get-config.
  + Also can update it on set-config

## Previous Releases

**Version: v0.0.19**

**Features/changes:**

* Change in Data storage service hosted VM IP

**Version: v0.0.18**

**Features/changes:**

* Upload Service is now working on periodic trigger instead of Redis Trigger

**Version: v0.0.17**

**Features/changes:**

* Included new version of AudioStream2py(0.1.21) library with bug fixes
* Added Session ID details to CLI when starting a capture

**Version: v0.0.16**

**Features/changes:**

* Included new version of AudioStream2py(0.1.20) library with bug fixes
* Fixed internal issue occurring when user starts a new session just after completing a session
* Created validation for not allowing users to create a new session when another session is already running

**Version: v0.0.15**

**Features/changes:**

* Removed chunk\_size filed and added chunk\_duration field in SS configuration file
* Added logs on CLI when DSS is down
* Changed CLI behavior as the user needs to wait till a block duration after pressed ctrl+c
* Updated Setup document with the steps for restarting DSS

**Version: v0.0.14**

**Features/changes:**

* Fixed DPP annotation upload issue occurring due to data size limitation in Cosmos DB
* Changed logger timezone to UTC

**Version: v0.0.13**

**Features/changes:**

* Improved logs on CLI on timer expiry and user stopping cases
* Added separate log files for message queue, services and CLI,
  + Message Queue -> msg\_queue\_oedge\_0.0.13.log
  + CLI -> cli\_oedge\_0.0.13.log
  + SS -> stream\_service\_oedge\_0.0.13.log
  + LSS -> local\_storage\_service\_oedge\_0.0.13.log
  + US -> upload\_service\_oedge\_0.0.13.log

**Version: v0.0.12**

**Features/changes:**

* Modified US to handle create session failure
* Modified US and LSS such that all messages go on single channel
* Fixed issue on CLI getting stuck on sessions with lesser duration

**Version: v0.0.11**

**Features/changes:**

* Added logs for LSS, SRS & AS.
* Added block count in logs.
* Fix for start-capture with -dpp option with and without timer.

**Version: v0.0.10**

**Features/changes:**

* DPP installation fix
* Better DSS logs

**Version: v0.0.9**

**Features/changes:**

* Added below logs:
  + In SRS SDI result.
  + CLI: Each user action and other info for debugging.
* Added log rotation feature with maximum file size 5 MB and maximum file count 50(Total 250 Mb) per services
* All service log files are combined in to single log file: **oedge\_0.0.9.log**
* Log files path changed to: /app/logs
  + Log files can be zipped and copied to the host machine. Please refer “**How to Get logs of Oedge**” section in setup document
* Config files path changed to: /app/configs

**Version: v0.0.8**

**Features/changes:**

* Added more logs for all the services. This can be used for analyzing gap issue in recordings
  + <https://app.asana.com/0/1203916667080587/1204747069758805>
* Log file names changed to service related names as follows,
  + Stream Service -> stream\_service.log
  + Local Storage Service -> local\_storage\_service.log
  + Upload Service -> upload\_service.log
* Added more logs to DSS server, to find the issue regarding accessibility

**Version: v0.0.7**

**Features/changes:**

* Fixed Gap issue in session audio file
* Fixed issue, where CLI cannot stop session on pressing Ctrl + C at some edge case
* Added new logs for all three services for specific case checks
* Made DSS running continuously on host, so that QA hasn’t need to take care of it

**Version: v0.0.6**

**Features/changes:**

* Removed autoselection of input device. Now the user needs to update the input device name in the config file. Steps for the same are explained in the Setup Document. By default, it will choose IEPE converters.
* Persisted the Configuration files in the host system. So, even if we rebuilt or deleted the container, the config changes will remain.
* Integrated new data storage service from platform\_poc with the device
* Config updation on stream service config file where done by CLI previously. Now this is changed and done by stream service itself.
* Session file saving path has been changed. See **Verify the Session file** section in the Setup Document.
* Azure MongoDB collection name changed from **session\_data\_collection** to **“sessions”**

**Version: v0.0.5**

**Features/changes:**

* Renamed services names as,
  + redis -> oedge\_redis
  + mongo -> oedge\_mongodb
  + other\_services -> oedge\_app

**Version: v0.0.4**

**Features/changes:**

* Used slabs in Analytics Service & Session Storage Service
* Changed service names
* Removed audio device index from config
* StreamService will check for IEPE, if not present it will use system default.
* If system default is not present or audio input does not work, StreamService will throw an error and exit.

**Version: v0.0.3**

**Features:**

* Added DPP annotations to Local and Azure MongoDB
* Added Autoselect IEPE converter feature

**Version: v0.0.2**

**Features:**

* Replaced .wav files with .pcm files
* Added DPP download option
* Can get dummy score from DPP if DPP enabled while starting capture

**Version: v0.0.1**

**Features:**

* Implemented end to end flow from CLI to MongoDB
* Flow
  + CLI -> Stream Service(Session Recording Service(SRS), Analytic Service(AS)) -> Local Storage Service -> Upload Service
  + CLI
    - CLI used to interact with DAQ
  + Stream Service
    - Creates two buffer readers for SRS and AS
    - SRS will create blocks from buffers and sent to local storage service
    - AS will create blocks from buffers and analyze it using DPP and sent a score to CLI
  + Local Storage Service
    - Save the block as a wav file locally and save session details in Mongodb. Also forward the data to Upload service
  + Upload Service
    - Save block on Azure file storage and CosmoDb using Data Storage API’s
  + Data Storage Service
    - Running on Azure VM.
* CLI is implemented as per following document
  + <https://docs.google.com/document/d/11Fiuu-3gx6DMeL4tYVsB9GHDAUEZsvra/edit?usp=share_link&ouid=113315534903626710685&rtpof=true&sd=true>