# ESB3003 Live Ingest Auto Configuration User Guide

| | |
|---|---|
| Document no | EDGS-235 |
| Version | A01 |
| Created on | 2025-06-12 |

CONFIDENTIAL

# Contents

# 1 Introduction

The live-ingest automatic channel configuration tool is designed to simpilfy the process of setting up and/or updating a channel. The automatic detection of the key information such PID, audio/video track(s), bitrate, language etc. from the transport stream minimizes the possibility of human error, ensuring reliable channel configurations. The tool produces a proposal for the channel configurations as a JSON file. The content of the file can be inspected and modified before being applied to the live ingest. The tool allows the user to perform analysis for multiple channels in parallel.

The tool's service can be configured to listen on a desired domain/IP-address and/or port via following environment variables.

- `AUTO_CONF_SERVER_ADDRESS` takes a string value while
- `AUTO_CONF_SERVER_ADDRESS` takes a port number.

# 2 Application Interfaces

The live-ingest automatic channel configuration tool offers a robust and flexible REST API for integration with other applications. The API enables the transport stream analysis and channel configurations via confd service. The REST API supports and utilizes JSON for the input data.

## 2.1 REST API Endpoints

- `POST /auto-conf/generate/channel` -- Initiate channel analysis
- `GET /auto-conf/generate/channel` -- Retrieve generated channel configurations report
- `GET /auto-conf/active/channel` -- Fetch live-ingest active channel configurations
- `POST /auto-conf/active/channel` -- Apply live-ingest channel configurations

### 2.1.1 POST /auto-conf/generate/channel -- Initiate channel analysis

This endpoint is used to initiate the channel transport stream analysis. It is possible to initiate up to ten analysis processes for the transport stream(s) of the channel(s) in parallel. The endpoint returns `202 ACCEPTED` if the input parameters are correct and the analysis has been started successfully.

```
POST /auto-conf/generate/channel

{
    "name": "string",
    "interface": "string",
    "inputSources": [
        {
            "address": "string",
            "igmpv3Source": "string",
            "port": 0
        }
    ],
    "timeout": 0
}
```

- **name:** The `name` is the channel name and it shall be unique for respective analysis process. This is a mandatory information.
- **interface:** The `interface` provides the network interface name; it is optional i.e. can be left empty.
- **inputSources:** The `inputSources` is a list of the channel input sources. The `address` can be a multicast address, a unicast address, a unicast address with SRT protocol or empty if not required. The IGMPv3 sourced multicast input sources are also supported; the `igmpv3Source` shall be set accordingly, otherwise it shall be empty. The `port` provides the port number where the channel transport stream is transmitted.
- **timeout:** The analysis can be time-boxed by setting `timeout` in seconds; it is set to 60 seconds by default.

### 2.1.2 GET /auto-conf/generate/channel -- Retrieve generated channel configurations report

This endpoint is used to retrieve the channel configurations report for the given channel. If the analysis for the channel transport stream is not completed yet, it will wait for the process to complete. The endpoint returns the generated channel configuration report, if successful. The channel configurations report complies with channel schema for `confd` and hence, it can be forwarded to `confd` service to configure a live-ingest channel.

The channel configurations report for a given channel can be requested as long as it is not overwritten by another channel analysis entry. The channel analysis entry is over- written either by the same channel name or after the maximum channel analysis queue i.e. ten.

Note that the tool does not check and resolve any conflicts of the generated channel configurations with active live-ingest configurations. The user is expected to check and resolve the conflicts.

```
GET /auto-conf/generate/channel?channelName=<name>
```

- **channelName:** The query parameter `channelName` corresponds to the `name` attribute in the data for `POST /auto-conf/generate/channel`.

```
{
    "channel": {
        ...
        "inputSources": [
            ...
        ],
        ...
        "name": "string",
        ...
    },
    "channelTemplate": {
        ...
        "name": "string",
        "tracks": [
            ...
        ],
    },
    "segmentationTemplate": {
        ...
        "name": "string",
        ...
    }
}
```

### 2.1.3 GET /auto-conf/active/channel -- Fetch live-ingest channel configurations

This endpoint is used to fetch active live-ingest channel configurations from `confd` service. The output result can be used to analyze and merge the generated channel configurations report in to the live-ingest channel configurations.

```
GET /auto-conf/active/channel?confdLocation=<confd-ip-address>&confdPort=<
    ↪ confd-port>
```

- **confdLocation:** The query parameter `confdLocation` provides the IP address to reach to "confd" service. It is required query parameter for the REST API to access intended `confd` service.
- **confdPort:** The query parameter `confdPort` provides the port number on which "confd" service is listening to incoming requests. If the query parameter is not provided, the default value 5000 is used.

#### 2.1.4 POST /auto-conf/active/channel -- Apply live-ingest channel configurations

This endpoint provides the support to apply a live-ingest channel configurations.

```
POST /auto-conf/active/channel?confdLocation=<confd-ip-address>&confdPort=<
    ↪ confd-port>&overwrite=<boolean>
```

- **confdLocation:** The query parameter `confdLocation` provides the IP address to reach to `confd` service. It is required query parameter for the REST API to access intended "confd" service.
- **confdPort:** The query parameter `confdPort` provides the port number on which `confd` service is listening to incoming requests. If the query parameter is not provided, the default value 5000 is used.
- **overwrite:** The boolean query parameter `overwrite` reflects if the live-ingest channel configurations shall be forced.