UTELISYS

Utelisys Cluster Transcoder API

Guide

Transcoder Developer Team

Version 1.0, November 2015

Table of Contents

The Utelisys Next Generation Transcoder family represents
the state-of-the-art for high-performance, video applications. The perfect platform for broadcast, mobile, cloud or OTT services, the Next Gen Transcoder, can scale and be configured to meet your specific requirements. Configured with both the latest highperformance, Digital Signal Processors (DSP) and General Purpose Processors (GPP), the hybrid architecture of the Utelisys platforms offer the best of all worlds, and are the optimal solution for the encoding, transcoding and distribution of your video streams.

Chapter 1. Overview

1.1. HTTP verbs

Verb	Desctiption
`GET`	Used to retrieve a resource
`POST`	Used to create a new resource
`PATCH`	Used for updating resources with partial JSON data. A PATCH request may accept one or more of the attributes to update the resource.
`DELETE`	Used to delete an existing resource

1.2. Client Errors

There are several possible types of client errors on API calls that receive request bodies

1. Sending invalid JSON results in a `400 Bad Request` response

   ```
   HTTP/1.1 400 Bad Request

   { "message":"Problems parsing JSON" }
   ```

2. Invalid authentication results in a `401 Unauthorized` response

   ```
   HTTP/1.1 401 Unauthorized

   { "message":"Bad credentials" }
   ```

3. Unauthorised access results in a `403 Forbidden` response

   ```
   HTTP/1.1 403 Forbidden

   {
     "timestamp":1427117816588,
     "status":403,
     "error":"Forbidden",
     "message":"Access Denied",
     "path":"/api/v1/users"
   }
   ```

4. Invalid values result in a `422 Unprocessable Entity` response

```
HTTP/1.1 422 Unprocessable Entity
```

```
{
  "message": "Validation failed",
  "errors": [
    { "field":"username", "message":"Username is required." },
    { "field": "password", "message":"Password is required." }
  ]
}
```

```
{
  "message": "A required parameter is missing",
  "errors": [
    {
      "field":"password",
      "message":"Required String parameter 'password' is not present"
    }
  ]
}
```

Chapter 2. Authentication

Authentication of API users

2.1. Login

Identifies and authenticates the user through the credentials presented by the user.

```
POST /api/v1/login
```

Input

Name	Type	Description
username	string	The user name of the user that wants to login
password	string	The password of the user that wants to login

Example

```
{ "username":"admin", "password":"blablabla" }
```

Success response

`200 OK`

```
{ "JSESSIONID":"3C043B9C7375DFC6E8003F075874E62F" }
```

Error response

`401 Unauthorized`

```
{ "message":"Bad credentials" }
```

Sample Call

```
curl -v \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-X POST -d '{ "username":"admin", "password":"blablabla" }' [HOST]/api/v1/login
```

2.2. Logout

Logs out the logged in user.

```
POST    /api/v1/logout
```

Data params

```
The authentication token "JSESSIONID" of the logged in user.
```

Sample Call

```
curl -v \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b "JSESSIONID=3C043B9C7375DFC6E8003F075874E62F" -X POST [HOST]/api/v1/logout
```

Chapter 3. Authorization

To do authorized request add cookie `JSESSIONID` that contains value obtained with Login response.

For example with curl tool use -c option to write all cookies in file and -b option to pass the data to the HTTP server as a cookie.

Sample call (using curl):

Authentication

```
curl \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-c auth -X POST -d '{ "username":"admin", "password":"blablabla" }'
[HOST]/api/v1/login
```

Authorized action

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X ....
```

Chapter 4. Users

4.1. Get a single user

```
GET /users/:id
```

Success response

```
{ "id":1, "username":"admin", "enabled":true, "role": "ADMIN" }
```

4.2. Get all users

```
GET /users
```

Success response

```
[
  { "id":1, "username":"admin", "enabled":true, "role": "ADMIN" },
  { "id":33, "username":"anonymus-1", "enabled":false, "role": "USER" },
  { "id":34, "username":"anonymus-2", "enabled":false, "role": "USER" },
  { "id":35, "username":"anonymus-3", "enabled":false, "role": "USER" }
]
```

4.3. Create user

```
POST /users
```

Data parameters

Name	Type	Required	Description
username	string (unique)	Yes	The name of the user
password	string	Yes	The password of the user in plain text
enabled	boolean	No	Access. (default false)
role	string (const)	Yes	User’s role. [**ADMIN** | **USER**]

Data example

```
{ "username":"anonymus", "password":"blablabla", "role": "USER" }
{ "username":"anonymus", "password":"blablabla", "enabled":true, "role": "USER" }
```

Success response

```
HTTP/1.1 200 OK
```

```
{ "id":35, "username":"anonymus-3", "enabled":false, "role": "USER" }
```

4.4. Update user

```
PATCH /users/:id
```

4.4.1. Change user name

Data parameters

Name	Type	Description
username	string (unique)	The new name of the user

Data example

```
{ "username":"master" }
```

Success response

```
HTTP/1.1 200 OK
```

```
{ "id":34, "username":"master", "enabled":false, "role": "USER" }
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message": "A user with this name already exists. Use a different name" }
```

4.4.2. Change password

Data parameters

Name	Type	Required	Description
oldPassword	string	Yes	The old password of the user in plain text
newPassword	string	Yes	The new password of the user in plain text

Data example

```
{ "oldPassword":"blablabla", "newPassword":"gogogo" }
```

Success response

```
HTTP/1.1 200 OK
```

```
{ "id":34, "username":"anonymus-2", "enabled":false, "role": "USER" }
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"The old password you entered was incorrect" }
```

4.4.3. Disable/Enable user

Data parameters

Name	Type	Description
enabled	boolean	Access

Data example

```
{ "enabled":false }
```

Success response

```
HTTP/1.1 200 OK
```

```
{ "id":34, "username":"anonymus-2", "enabled":false, "role": "USER" }
```

Chapter 5. Nodes

5.1. Get single node

```
GET /nodes/:id
```

Success response

```
HTTP/1.1 200 OK
```

```
{
    "id":1,
    "type":"TRANSCODING",
    "name":"S2 streaming",
    "description":"",
    "group":1,
    "resources":{
        "system":{ "totalram":5988790272, "nprocs":4 },
        "dsp_boards":[
            { "name":"DSP #0", "global_dsp_ids":[0,1,2,3] }
        ]
    },
    "host":"217.117.226.215",
    "status":"RUNNING",
    "errorMessage":""
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Node not found" }
```

5.2. List nodes

```
GET /nodes
```

Success response

```
HTTP/1.1 200 OK
```

```
[
    {
        "id":1,
        "type":"TRANSCODING",
        "name":"S2 streaming",
        "description":"",
        "group":1,
        "resources":{
            "system":{ "totalram":5988790272, "nprocs":4 },
            "dsp_boards":[
                { "name":"DSP #0", "global_dsp_ids":[0,1,2,3] }
            ]
        },
        "host":"217.117.226.215",
        "status":"RUNNING",
        "errorMessage":""
    },
    {
        "id":2,
        "type":"STREAMING",
        "name":"S2 streaming",
        "description":"",
        "group":2,
        "resources":{},
        "host":"217.117.226.215",
        "status":"RUNNING",
        "errorMessage":""
    }
]
```

5.3. Create a node

```
POST /nodes
```

Data parameters

Name	Type	Required	Description
name	string	Yes	Human readable node name
type	string (const)	Yes	Node type. [**TRANSCODING** | **STREAMING**]
description	string	No	A description of the node
group	number	Yes	The **node group** that this node is a member of

Name	Type	Required	Description
host	string	Yes	The ... Can be IPv4, IPv6, FQDN
status	string (const)	Yes	Node status. [**NEW** | **DISABLED**]

Data example

```
{
    "name": "Node1",
    "description": "Some test node",
    "type": "TRANSCODING",
    "status": "NEW",
    "host": "127.0.0.1",
    "group": 1
}
```

Success response

```
HTTP/1.1 200 OK
```

```
{
    "id":4,
    "type":"TRANSCODING",
    "name":"Node1",
    "description":"Some test node",
    "group":1,
    "resources":null,
    "host":"127.0.0.1",
    "status":"NEW",
    "errorMessage":null
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{
    "message":"Validation failed",
    "errors":[
        { "field":"group", "message":"Node group not found." }
    ]
}
```

If node and node group has different type

```
{
    "message":"Validation failed",
    "errors":[
        { "field":"group", "message":"Invalid node group" }
    ]
}
```

```
{
    "message":"Validation failed",
    "errors":[
        { "field":"host", "message":"The host address is not valid" }
    ]
}
```

```
{
    "message":"Validation failed",
    "errors":[
        { "field":"status", "message":"Node status is not permitted" }
    ]
}
```

5.4. Edit a node

```
PATCH /nodes/:id
```

Data parameters

Name	Type	Description
name	string	Human readable node name
description	string	A description of the node

Name	Type	Description
group	number	The **node group** that this node is a member of
host	string	The … Can be IPv4, IPv6, FQDN
status	string (const)	Node status. [**NEW** | **DISABLED**]

Example 1. Rename node and description

```
curl -s -i \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "name": "New node name", "description": "" }'
[HOST]/api/v1/nodes/1
```

Example 2. Change node host

```
curl -s -i \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "host": "s4.utelisys.tv" }' [HOST]/api/v1/nodes/1
```

⚠ The node should have **DISABLED** *status* before *host* change.

Example 3. Move **node** *to another* **node group**

```
curl -s -i \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "group": 3 }' [HOST]/api/v1/nodes/1
```

⚠ **Node group** type should be the same as **node** type

5.5. Remove a node

```
DELETE /nodes/:id
```

Success response

```
HTTP/1.1 200 OK
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Node not found." }
```

```
{ "message":"This node is referenced by or running tasks. It can't be deleted
right now." }
```

Chapter 6. Node groups

6.1. Get single node group

```
GET /nodegroups/:id
```

Success response

```
HTTP/1.1 200 OK
```

```
{
  "id":1,
  "type":"TRANSCODING",
  "name":"Main transcoding group",
  "description":"",
  "transcodingDestinationTemplate":"udp://239.100.100.T[0-255]:S[4000-4100]",
  "nodes":[1]
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Node group not found." }
```

6.2. List node groups

```
GET /nodegroups
```

Success response

```
HTTP/1.1 200 OK
```

```
[
  {
    "id":1,
    "type":"TRANSCODING",
    "name":"Main transcoding group",
    "description":"",
    "transcodingDestinationTemplate":"udp://239.100.100.T[0-255]:S[4000-4100]",
    "nodes":[1]
  },
  {
    "id":2,
    "type":"STREAMING",
    "name":"Main streaming group",
    "description":"",
    "transcodingDestinationTemplate":"",
    "nodes":[2]
  }
]
```

6.3. Create a node group

```
POST /nodegroups
```

Data parameters

Name	Type	Required	Description
name	string	Yes	Human readable node group name
type	string (const)	Yes	Node group type. [TRANSCODING | STREAMING]
description	string	No	A description of the node group.

Data example

```
{
  "name": "Office transcoding group",
  "description": "Transcoding group in office",
  "type": "TRANSCODING"
}
```

Success response

HTTP/1.1 200 OK

```
{
  "id":5,
  "type":"TRANSCODING",
  "name":"Office transcoding group",
  "description":"Transcoding group in office",
  "transcodingDestinationTemplate":null,
  "nodes":[]
}
```

Error response

HTTP/1.1 400 Bad Request

```
{
  "message":"Validation failed",
  "errors":[
    {"field":"name","message":"Node group name is required"},
    {"field":"type","message":"Node group type is required"}
  ]
}
```

```
{
  "message":"Validation failed",
  "errors":[
    {"field":"type","message":"Invalid format"}
  ]
}
```

6.4. Edit a node group

```
PATCH /nodegroups/:id
```

Data parameters

Name	Type	Description
name	string	Human readable node group name
description	string	A description of the node group

Success response

```
HTTP/1.1 200 OK
```

```
{
  "id":7,
  "type":"TRANSCODING",
  "name":"Bla-bla-bla",
  "description":"Transcoding group in office",
  "transcodingDestinationTemplate":null,
  "nodes":[]
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{
  "message":"Validation failed",
  "errors":[
    {"field":"type","message":"The field is read only"}
  ]
}
```

6.5. Remove a node group

```
DELETE /nodegroups/:id
```

Success response

```
HTTP/1.1 200 OK
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Node group not found." }
```

```
{ "message":"This node group is referenced by node. It can't be deleted right
now." }
```

Chapter 7. Codec profiles

7.1. Get codec profle

```
GET /codecprofiles/:id
```

Success response

```
HTTP/1.1 200 OK
```

```
{
  "id":4,
  "name":"AAC (default)",
  "description":"Some codec paramiters",
  "techDescription":"Stereo, 48 khz, 196 kbit",
  "type":"Audio",
  "codec":"AAC",
  "parameters":{
    "bit_rate":196000,
    "sample_rate":48000,
    "channels":2
  }
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Codec profile not found" }
```

7.2. List of codec profles

```
GET /codecprofiles
```

Success response

```
HTTP/1.1 200 OK
```

```
[
  {
      "id":1,
      "name":"H.264 (default)",
      "description":"",
      "techDescription":"HP, Level 4.0, 8000 kbit",
      "type":"Video",
      "codec":"TI H.264",
      "parameters":{
        "CbQPIndexOffset": 0,
        "CrQPIndexOffset": 0,
        "HRDBufferSize": 8000000,
        "IDRFrameInterval": 1,
        "MvRangeHorB": 144,
        "MvRangeHorP": 144,
        "MvRangeVerB": 32,
        "MvRangeVerP": 32,
        "aspectRatioIdc": 255,
        "aspectRatioInfoPresentFlag": 1,
        "bitRate": 8000000,
        "colourDescriptionPresentFlag": 1,
        "colourPrimaries": 5,
        "constrainedIntraPredEnable": 0,
        "enableIntraPartition": 1,
        "enablePRC": 1,
        "encodingPreset": 3,
        "entropyCodingMode": 0,
        "filterOffsetA": 0,
        "filterOffsetB": 0,
        "frameSkipAfterSceneChange": 1,
        "gopStructure": 0,
        "initialBufferLevel": 8000000,
        "interCodingPreset": 1,
        "interFrameInterval": 1,
        "interlaceCodingType": 4,
        "intraCodingPreset": 1,
        "intraFrameInterval": 24,
        "intraRefreshMethod": 1,
        "intraRefreshRate": 0,
        "level": 40,
        "log2MaxFNumMinus4": 10,
        "loopfilterDisableIDC": 2,
        "loopfilterPreset": 1,
        "matrixCoefficients": 5,
        "maxBitRate": 16000000,
        "maxInterFrameInterval": 1,
        "maxMVperMB": 1,
        "minBitRate": 8000000,
        "mvAccuracy": 0,
        "picOrderCountType": 0,
        "profile": 100,
        "qpI": -1,
        "qpMaxB": 35,
      "qpMaxI": 35,
      "qpMaxP": 35,
      "qpMinB": 10,
      "qpMinI": 10,
      "qpMinP": 10,
      "qpOffsetB": 4,
      "qpP": -1,
      "rateControlParamsPreset": 1,
      "rateControlPreset": 5,
      "rcAlgo": 0,
      "sampleAspectRatioHeight": 0,
      "sampleAspectRatioWidth": 0,
      "scalingMatrixPreset": 1,
      "sliceCodingPreset": 0,
      "sliceMode": 0,
      "sliceUnitSize": 0,
      "streamFormat": 0,
      "timingInfoPresentFlag": 1,
      "transferCharacteristics": 5,
      "transformBlockSize": 0,
      "videoFormat": 2,
      "videoFullRangeFlag": 0,
      "videoSignalTypePresentFlag": 0,
      "vuiCodingPreset": 1
    }
  },
  {
    "id":2,
    "name":"AAC (default)",
    "description":"",
    "techDescription":"",
    "type":"Audio",
    "codec":"AAC",
    "parameters":{
      "bit_rate":-1,
      "sample_rate":-1,
      "channels":2
    }
  },
  {
    "id":3,
    "name":"AAC (default)",
    "description":"",
    "techDescription":"",
    "type":"Audio",
    "codec":"AAC",
    "parameters":{
      "bit_rate":-1,
      "sample_rate":-1,
      "channels":5
    }
  },
  {
    "id":4,
    "name":"AAC (default)",
    "description":"Some codec paramiters",
    "techDescription":"Stereo, 48 khz, 196 kbit",
    "type":"Audio",
    "codec":"AAC",
    "parameters":{
      "bit_rate":196000,
      "sample_rate":48000,
      "channels":2
    }
  }
]
```

7.3. Create a codec profile

```
POST /codecprofiles
```

Data parameters

Name	Type	Required	Description
name	string	Yes	The codec profile name
description	string	No	The description of the codec profile
codec	string (const)	Yes	Codec name. [`TI H.264`, `AAC`]
parameters	object	Yes	Parameters, specific for codec

Supported codecs

- `TI H.264` (properties)
- `AAC` (properties)

Success response

```
HTTP/1.1 200 OK
```

```
{
  "id":36,
  "name":"AAC (default)",
  "description":"Some codec parameters",
  "techDescription":"Stereo, 48 khz, 196 kbit",
  "type":"Audio",
  "codec":"AAC",
  "parameters":{
    "bit_rate":196000,
    "sample_rate":48000,
    "channels":2
  }
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{
  "message":"Validation failed",
  "errors":[
    { "field":"name", "message":"Profile name is required" }
  ]
}
```

`HTTP/1.1 422 Unprocessable Entity`

```
{
  "message":"Validation failed",
  "errors":[
    { "field":"codec", "message":"Invalid codec name" }
  ]
}
```

```
{
  "message":"Validation failed",
  "errors":[
    { "field":"parameters.bit_rate", "message":"You must specify the property" },
    { "field":"parameters.sample_rate", "message":"You must specify the property"
},
    { "field":"parameters.channels", "message":"You must specify the property" }
  ]
}
```

Example 4. Create audio codec

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X POST -d @codec_1.json [HOST]/api/v1/codecprofiles
```

codec_1.json

```
{
  "name": "AAC (default)",
  "description": "Some codec parameters",
  "codec": "AAC",
  "parameters": {
    "bit_rate": 196000,
    "sample_rate": 48000,
    "channels": 2
  }
}
```

Example 5. Create video codec

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X POST -d @codec_2.json [HOST]/api/v1/codecprofiles
```

codec_2.json

```
{
  "name": "H.264 (default)",
  "codec": "TI H.264",
  "parameters": {
    "interCodingPreset":1,
    "videoFullRangeFlag":0,
    "qpMaxI":35,
    "sliceCodingPreset":0,
    "qpMaxP":35,
    "qpI":-1,
    "HRDBufferSize":8000000,
    "loopfilterPreset":1,
    "qpP":-1,
    "aspectRatioInfoPresentFlag":1,
    "intraCodingPreset":1,
    "transformBlockSize":0,
    "rateControlParamsPreset":1,
    "sampleAspectRatioHeight":0,
    "vuiCodingPreset":1,
    "gopStructure":0,
    "sliceMode":0,
    "colourPrimaries":5,
    "aspectRatioIdc":255,
    "maxMVperMB":1,
    "interFrameInterval":1,
    "initialBufferLevel":8000000,
    "matrixCoefficients":5,
    "level":40,
    "profile":100,
    "MvRangeHorB":144,
    "colourDescriptionPresentFlag":1,
    "entropyCodingMode":0,
    "timingInfoPresentFlag":1,
    "intraRefreshMethod":1,
    "MvRangeHorP":144,
    "rcAlgo":0,
    "CrQPIndexOffset":0,
    "sliceUnitSize":0,
    "qpMaxB":35,
    "sampleAspectRatioWidth":0,
    "picOrderCountType":0,
    "enablePRC":1,
    "qpMinB":10,
    "maxInterFrameInterval":1,
    "scalingMatrixPreset":1,
    "qpMinI":10,
    "bitRate":8000000,
    "qpOffsetB":4,
    "interlaceCodingType":4,
    "enableIntraPartition":1,
    "streamFormat":0,
    "maxBitRate":16000000,
    "mvAccuracy":0,
    "IDRFrameInterval":1,
    "encodingPreset":3,
    "MvRangeVerB":32,
    "loopfilterDisableIDC":2,
    "transferCharacteristics":5,
    "qpMinP":10,
    "intraRefreshRate":0,
    "videoFormat":2,
    "MvRangeVerP":32,
    "CbQPIndexOffset":0,
    "frameSkipAfterSceneChange":1,
    "filterOffsetB":0,
    "filterOffsetA":0,
    "minBitRate":8000000,
    "log2MaxFNumMinus4":10,
    "videoSignalTypePresentFlag":0,
    "intraFrameInterval":24,
    "rateControlPreset":5,
    "constrainedIntraPredEnable":0
  },
  "description": "TI H.264 codec bla-bla-bla "
}
```

7.4. Edit a codec profile

```
PATCH /codecprofiles/:id
```

Data parameters

Name	Type	Description
name	string	The codec profile name
description	string	The description of the codec profile
parameters	object	Parameters, specific for codec

Success response

```
HTTP/1.1 200 OK
```

```
{
  "id":2,
  "name":"New name for AAC",
  "description":"",
  "techDescription":"Stereo, 48 khz, 196 kbit",
  "type":"Audio",
  "codec":"AAC",
  "parameters":{
    "bit_rate":196000,
    "sample_rate":48000,
    "channels":2
  }
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{
  "message":"Validation failed",
  "errors":[
    { "field":"name", "message":"Profile name is required" }
  ]
}
```

```
HTTP/1.1 422 Unprocessable Entity
```

```
{
  "message":"Validation failed",
  "errors":[
    {"field":"parameters._error_0","message":"Bit Rate must be in range from Min
Bit Rate to Max Bit Rate inclusive."}
  ]
}
```

Example 6. Rename codec

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "name": "New name for AAC codec" }'
[HOST]/api/v1/codecprofiles/2
```

Example 7. Change codec parameters

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "parameters": { "channels": 5 } }'
[HOST]/api/v1/codecprofiles/2
```

7.5. Remove a codec profile

```
DELETE /codecprofiles/:id
```

Success response

```
HTTP/1.1 200 OK
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message": "Codec profile not found" }
```

```
{ "message": "The operation could not be completed because the item is used" }
```

Chapter 8. Tasks

8.1. Get single task

```
GET /tasks/:id
```

Success response

```
HTTP/1.1 200 OK
```

```
{
  "id":1,
  "name":"SDI",
  "description":"",
  "state":"RUNNING",
  "errors":[],
  "sourceType":"SDI",
  "sourceParameters":{
    "type":"SDI",
    "boardId":0,
    "inputId":0
  },
  "transcodingNodeGroup":1,
  "transcodingNode":1,
  "streamingNodeGroup":2,
  "streamingNode":null,
  "autorestart":false,
  "programs":[
    {
      "id":1,
      "pid":null,
      "serviceName":null,
      "serviceProvider":null,
      "transcodings":[
        {
          "id":1,
          "destination":"udp://239.100.100.3:4000",
          "streams":[
            {
              "id":1,
              "description":"1920:1080p 25 fps",
              "type":"Video",
              "index":0,
              "mediaAttributes":{
                "type":"Video",
                "maxWidth":null,
                "maxHeight":null,
                "frameScanType":"Progressive",
                "frameRate":null
              },
              "codecProfile":1,
              "skip":false,
              "passthrough":false,
              "hints":{}
            },
            {
              "id":2,
              "description":"",
              "type":"Audio",
              "index":0,
              "mediaAttributes":{ "type":"Audio" },
              "codecProfile":2,
              "skip":false,
              "passthrough":false,
              "hints":{}
            },
            {
              "id":3,
              "description":"",
              "type":"Audio",
              "index":1,
              "mediaAttributes":{ "type":"Audio" },
              "codecProfile":3,
              "skip":false,
              "passthrough":false,
              "hints":{}
            }
          ]
        },
        {
          "id":2,
          "destination":"udp://239.100.100.3:4001",
          "streams":[
            {
              "id":4,
              "description":"1280:720p 25 fps",
              "type":"Video",
              "index":0,
              "mediaAttributes":{
                "type":"Video",
                "maxWidth":1280,
                "maxHeight":720,
                "frameScanType":"Progressive",
                "frameRate":null
              },
              "codecProfile":1,
              "skip":false,
              "passthrough":false,
              "hints":{}
            },
            {
              "id":5,
              "description":"",
              "type":"Audio",
              "index":0,
              "mediaAttributes":{ "type":"Audio" },
              "codecProfile":2,
              "skip":false,
              "passthrough":false,
              "hints":{}
            },
            {
              "id":6,
              "description":"",
              "type":"Audio",
              "index":1,
              "mediaAttributes":{ "type":"Audio" },
              "codecProfile":3,
              "skip":false,
              "passthrough":false,
              "hints":{}
            }
          ]
        },
        {
          "id":3,
          "destination":"udp://239.100.100.3:4002",
          "streams":[
            {
              "id":7,
              "description":"960:540p 25 fps",
              "type":"Video",
              "index":0,
              "mediaAttributes":{
                "type":"Video",
                "maxWidth":960,
                "maxHeight":540,
                "frameScanType":"Progressive",
                "frameRate":null
              },
              "codecProfile":1,
              "skip":false,
              "passthrough":false,
              "hints":{}
            },
            {
              "id":8,
              "description":"",
              "type":"Audio",
              "index":0,
              "mediaAttributes":{ "type":"Audio" },
              "codecProfile":2,
              "skip":false,
              "passthrough":false,
              "hints":{}
            },
            {
              "id":9,
              "description":"",
              "type":"Audio",
              "index":1,
              "mediaAttributes":{ "type":"Audio" },
              "codecProfile":3,
              "skip":false,
              "passthrough":false,
              "hints":{}
            }
          ]
        }
      ],
      "streamings":[
        {
          "id":1,
          "type":"HLS",
          "outcomeStreams":[
            {
              "type":"HLS",
              "uri":
"http://185.59.37.129/f67c3da8a917d607e9f66d9788cfff83670dae44/playlist.m3u8",
              "substreams":[
                {
                  "type":"HLS",
                  "uri":
"http://185.59.37.129/f67c3da8a917d607e9f66d9788cfff83670dae44/ef6464030fa0/playli
st.m3u8",
                  "bandwidth":8324000
                },
                {
                  "type":"HLS",
                  "uri":
"http://185.59.37.129/f67c3da8a917d607e9f66d9788cfff83670dae44/ef6464030fa1/playli
st.m3u8",
                  "bandwidth":8324000
                },
                {
                  "type":"HLS",
                  "uri":
"http://185.59.37.129/f67c3da8a917d607e9f66d9788cfff83670dae44/ef6464030fa2/playli
st.m3u8",
                  "bandwidth":8324000
                }
              ]
            }
          ],
          "transcodingIndexes":[0,1,2],
          "settings":null
        }
      ],
      "transcodingHints":{ "decoding":"DSP" },
      "streamingHints":{ },
      "streams":[
        {
          "type":"Video",
          "index":0,
          "description":"1920:1080p 25 fps",
          "language":null,
          "mediaAttributes":{
            "type":"Video",
            "maxWidth":1920,
            "maxHeight":1080,
            "frameScanType":"Progressive",
            "frameRate":25
          },
          "videoStandard":null,
          "vbiDigitalizeBitrate":null,
          "channels":null,
          "overrideFrameRate":null
        },
        {
          "type":"Audio",
          "index":0,
          "description":"Channels: 1, 2",
          "language":null,
          "mediaAttributes":{ "type":"Audio" },
          "videoStandard":null,
          "vbiDigitalizeBitrate":null,
          "channels":[0,1],
          "overrideFrameRate":null
        },
        {
          "type":"Audio",
          "index":1,
          "description":"Channels: 3, 4, 5, 6, 7",
          "language":null,
          "mediaAttributes":{ "type":"Audio" },
          "videoStandard":null,
          "vbiDigitalizeBitrate":null,
          "channels":[2,3,4,5,6],
          "overrideFrameRate":null
        }
      ]
```

```
    }
  ]
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Task not found" }
```

8.2. List tasks

```
GET /tasks
```

8.3. Create a task

```
POST /tasks
```

Data parameters

Name	Type	Required	Description
name	string	Yes	The name of the task
description	string	No	The description of the task
state	string (const)	Yes	Indicates the state of the task to create. [**STOPPED** | **SCHEDULED_TO_RUN** | **SCHEDULED_TO_STOP**]
sourceType	string (const)	Yes	The type of source [**UDP** | **SDI**]
sourceParameters	object	Yes	The parameters of source (**SDI**, **UDP**)
transcodingNodeGroup	number	No	The ID of nodes group, that will be used for transcoding
transcodingNode	number	Yes/No	The node ID that will be used for transcoding. Required for **SDI** source.
streamingNodeGroup	number	No	The ID of nodes group, that will be used for streaming
streamingNode	number	No	The node ID that will be used for streaming

Name	Type	Required	Description
autorestart	boolean	No	Enable auto restart task error or node unavailability
programs	array of object	Yes	A list of programs (Program)

Table 1. SDI source parameters

Name	Type	Required	Description
type	string (const)	Yes	**SDI**
boardId	number	Yes	The SDI board ID in system
inputId	number	Yes	The SDI input ID on board

Table 2. UDP source parameters

Name	Type	Required	Description
type	string (const)	Yes	**UDP**
source	string	Yes	The source URI

Table 3. Program

Name	Type	Required	Description
pid	number	No	The program ID
serviceName	string	No	The service name
serviceProvider	string	No	The service provider
sourceStreams	array of object	Yes	A list of source stream for current program (Source stream)
transcodings	array of object	Yes	A list of transcodings (Transcoding)
streamings	array of object	No	A list of streamings (Streaming)
transcodingHints	array of object	No	A list of additional options for transcoding (array of key/value objects) Transcoding hints
streamingHints	array of object	No	A list of Additional options for streaming (array of key/value objects) Streaming hints

Table 4. Transcoding hints

Name	Type	Description
decoding	string (const)	Video decoding on [**HOST** | **DSP**] **HOST** - decoding on host by ffmpeg, **DSP** - decoding on DSP board by TI codec

Table 5. Streaming hints

Name	Type	Description
passthrough_udpxy	boolean	Use udpxy between **transcoding node** and **streaming node**

Table 6. Source stream

Name	Type	Required	Description
type	string (const)	Yes	The media type [**Video** | **Audio** | **Subtitle**]
index	number	Yes	The stream index [0 to n]. Stream is indexing separately for each media type.
mediaAttributes	object	Yes	Media attributes (Video, Audio, Subtitle)
videoStandard	number	No	Overrides autodetected video standard for specified SDI input (SDI standards)
vbiDigitalizeBitrate	number	No	Overrides vbi digitalize bitrate for specified SDI input
channels	array of number	No	A list of audio channel indexes (for SDI audio streams)
overrideFrameRate	number	No	Overrides frame rate (for video streams)

Table 7. Source stream **video** *media attributes*

Name	Type	Required	Description
type	string (const)	Yes	**Video**
maxWidth	number	Yes	Maximum width of frame
maxHeight	number	Yes	Maximum height of frame
frameScanType	string (const)	Yes	[**Dynamic** | **Progressive** | **Interlaced**]
frameRate	number	Yes	Frame rate

Table 8. Source stream **audio** *media attributes*

Name	Type	Required	Description
type	string (const)	Yes	**Audio**

Table 9. Source stream **subtitle** *media attributes*

Name	Type	Required	Description
type	string (const)	Yes	**Subtitle**

Table 10. Transcoding

Name	Type	Required	Description
destination	string	Yes	The destination of transcoding

Name	Type	Required	Description
streams	array of object	Yes	List of streams to be transcoding Stream

Table 11. Stream

Name	Type	Required	Description
type	string (const)	Yes	The media type [**Video** | **Audio** | **Subtitle**]
index	number	Yes	The stream index [0 to n]. Stream is indexing separately for each media type. The index should be corresponded to source stream index.
mediaAttributes	object	No	
codecProfile	number	No	The ID of codec profile to be used for encoding
skip	boolean	No	Do not process this stream
passthrough	boolean	No	Pass this stream without changes
hints	array of object	No	A list of additional options for stream (array of key/value objects) Stream hints

⚠ One of the properties: **codecProfile**, **skip**, **passthrough**, should be set

💡 The properties that specified in **mediaAttributes** will lead changes in stream. The **maxWidth** and **maxHeight** will activate resizing. The **frameScanType** manipulates with deiterlacer.

Table 12. Stream hints

Name	Type	Required	Description
core_count	number	No	Number of DSP cores for encoding
core_usage	number	No	DSP encoder usage per core

Table 13. Streaming

Name	Type	Required	Description
type	string (const)	Yes	Streaming type [**HLS** | **HDS** | **MSS** | **DASH**]
transcodingIndexes	array of number	Yes	A list of transcoding indexes
settings	object	No	The additional streaming settings (Streaming settings)

Table 14. Streaming settings

Name	Type	Required	Description
chunkLength	number	No	The length (in seconds) of fragments to be made
chunkCount	number	No	The number of fragments before the server starts to overwrite the older fragments

8.3.1. Task structure description

Source settings

Listing 1. SDI

```
{
    . . .
    "sourceType": "SDI",     ①
    "sourceParameters": {
        "type": "SDI",
        "boardId": 0,          ②
        "inputId": 0           ③
    }
    . . .
}
```

① Type of source

② The board ID in system

③ The input ID on board

Listing 2. UDP

```
{
    . . .
    "sourceType": "UDP",     ①
    "sourceParameters": {
        "type": "UDP",
        "source": "udp://239.100.7.1:4422"   ②
    }
    . . .
}
```

① Type of source

② Source in network (URI)

Source streams

Listing 3. SDI

```
{
    . . .
    "programs": [     ①
        {
            . . .
            "streams": [     ②
                {
                    "type": "Video",     ③
                    "index": 0,          ③
                    "mediaAttributes": {         ④
                        "type": "Video",
                        "maxWidth": 1920,                   ⑤
                        "maxHeight": 1080,                  ⑤
                        "frameScanType": "Progressive",     ⑥
                        "frameRate": 25                     ⑦
                    },
                    "videoStandard": null,            ⑧
                    "vbiDigitalizeBitrate": null,     ⑨
                },
                {
                    "type": "Audio",     ③
                    "index": 0,          ③
                    "mediaAttributes": { "type": "Audio" },     ④
                    "channels": [0, 1]                   ⑩
                },
                {
                    "type": "Audio",     ③
                    "index": 1,          ③
                    "mediaAttributes": { "type": "Audio" },     ④
                    "channels": [ 2, 3, 4, 5, 6]         ⑩
                }
            ]
        }
    ]
}
```

① Each task may has several programs (at least one)

② The program has a list of source streams

③ Stream type and index [0 to n]. Stream is indexing separately for each media type.

④ Each stream has media attributes

⑤ Video frame resolution

⑥ Video frame scan type (**Progressive**, **Interlaced**, **Dynamic**)

⑦ Video stream frame rate

⑧ Overrides autodetected video standard for specified SDI input (Deltacast SDI standards)

⑨ Overrides vbi digitalize bitrate for specified SDI input

⑩ A list of audio channel indexes

Listing 4. UDP

```
{
    . . .
    "programs": [     ①
        {
            . . .
            "streams": [         ②
                {
                    "type": "Video",             ③
                    "index": 0,                  ③
                    "mediaAttributes": {                 ④
                        "type": "Video",
                        "maxWidth": 720,                        ⑤
                        "maxHeight": 576,                       ⑤
                        "frameScanType": "Dynamic",             ⑥
                        "frameRate": 25                         ⑦
                    },
                    "overrideFrameRate": null             ⑧
                },
                {
                    "type": "Audio",             ③
                    "index": 0,                  ③
                    "mediaAttributes": { "type": "Audio" }             ④
                },
                {
                    "type": "Audio",             ③
                    "index": 1,                  ③
                    "mediaAttributes": { "type": "Audio" }             ④
                },
                {
                    "type": "Audio",             ③
                    "index": 2,                  ③
                    "mediaAttributes": { "type": "Audio" }             ④
                },
                {
                    "type": "Subtitle",             ③
                    "index": 0,                     ③
                    "mediaAttributes": { "type": "Subtitle" }             ④
                },
                {
                    "type": "Subtitle",             ③
                    "index": 1,                     ③
                    "mediaAttributes": {  "type": "Subtitle" }             ④
                }
            ]
        }
    ]
}
```

① Each task may has several programs (at least one)

② The program has a list of source streams

③ Stream type and index [0 to n]. Stream is indexing separately for each media type.

④ Each stream has media attributes

⑤ Video frame resolution

⑥ Video frame scan type (**Progressive**, **Interlaced**, **Dynamic**)

⑦ Video stream frame rate

⑧ Overrides autodetected frame rate for video stream

Transcodings

```
{
    . . .
    "programs": [     ①
        {
            "transcodings": [     ②
                {
                    "destination": "udp://239.100.100.2:4000",     ③
                    "streams": [     ④
                        {
                            "type": "Video",     ⑤
                            "index": 0,          ⑤
                            "mediaAttributes": {        ⑥
                                "type": "Video",
                                "maxWidth": null,
                                "maxHeight": null,
                                "frameScanType": "Progressive",     ⑦
                                "frameRate": null
                            },
                            "codecProfile": 1,       ⑧
                            "skip": false,           ⑨
                            "passthrough": false,    ⑩
                            "hints": { }        ⑪
                        },
                        {
                            "type": "Audio",     ⑤
                            "index": 0,          ⑤
                            "mediaAttributes": { "type": "Audio" },     ⑥
                            "codecProfile": 2,       ⑧
                            "skip": false,           ⑨
                            "passthrough": false,    ⑩
                            "hints": {  }        ⑪
                        },
                        {
                            "type": "Audio",     ⑤
                            "index": 1,          ⑤
                            "mediaAttributes": { "type": "Audio" },         ⑥
                            "codecProfile": 3,       ⑧
                            "skip": false,            ⑨
                            "passthrough": false,     ⑩
                            "hints": { }         ⑪
                        }
                    ]
                },
                {
                    "destination": "udp://239.100.100.2:4001",     ③
                    "streams": [{     ④
                        "type": "Video",     ⑤
                        "index": 0,          ⑤
                        "mediaAttributes": {         ⑥
                            "type": "Video",
                            "maxWidth": 1280,                   ⑫
                            "maxHeight": 720,                   ⑫
                            "frameScanType": "Progressive",     ⑦
                            "frameRate": null
                        },
                        "codecProfile": 1,        ⑧
                        "skip": false,            ⑨
                        "passthrough": false,     ⑩
                        "hints": { }         ⑪
                    },
                    {
                        "type": "Audio",     ⑤
                        "index": 0,          ⑤
                        "mediaAttributes": { "type": "Audio" },         ⑥
                        "codecProfile": 2,        ⑧
                        "skip": false,            ⑨
                        "passthrough": false,     ⑩
                        "hints": { }         ⑪
                    },
                    {
                        "type": "Audio",     ⑤
                        "index": 1,          ⑤
                        "mediaAttributes": { "type": "Audio" },         ⑥
                        "codecProfile": 3,        ⑧
                        "skip": false,            ⑨
                        "passthrough": false,     ⑩
                        "hints": { }         ⑪
                    }
                ]
                },
                {
                    "destination": "udp://239.100.100.2:4002",     ③
                    "streams": [     ④
                        . . .
                    ]
                }
            ]
        }
```

```
    ]
}
```

① Each task may has several programs (at least one)

② The program has a list of transcodings

③ Transcoding has destination, where transcoded streams will be sended after muxing

④ Each transcoding has list of streams that will be transformed and muxed

⑤ The stream type and index [0 to n]. Streams are indexing separately for each media type. The index should be corresponded to source stream index.

⑥ media attributes

⑦ Enables deinterlaser, if source video stream is interlaced

⑧ the codec profile id

⑨ Do not process this stream

⑩ Pass this stream without changes

⑪ Transcoding hints

⑫ Enables resizing input video stream to specified resolution

Streamings

```
{
    . . .
    "programs": [    ①
        {
            . . .
            "streamings": [    ②
                {
                    "type": "HLS",    ③
                    "transcodingIndexes": [ 0, 1, 2 ],    ④
                    "settings": null    ⑤
                }
            ]
        }
    ]
}
```

① Each task may has several programs (at least one)

② The program has a list of streaming

③ Type of produced stream

④ A list of transcoding indexes

⑤ The additional streaming settings (Streaming settings)

8.4. Start/Stop task

```
PATCH /tasks/:id
```

Data parameters

Name	Type	Description
state	string (const)	Task state. [**STOPPED** | **SCHEDULED_TO_RUN** | **SCHEDULED_TO_STOP**]

Look at task state diagram to recognize task workflow. The bold arrows show available actions for user.

To run or stop task you should manipulate with task states.

Example 8. Run task

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "state":"SCHEDULED_TO_RUN" }' [HOST]/api/v1/tasks/1
```

Example 9. Stop task

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "state":"SCHEDULED_TO_STOP" }' [HOST]/api/v1/tasks/1
```

Example 10. Clear error state

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X PATCH -d '{ "state":"STOPPED" }' [HOST]/api/v1/tasks/1
```

Success response

```
HTTP/1.1 200 OK
```

```
{
    "id":1,
    "name":"Task #1",
    "description":"",
    "state":"SCHEDULED_TO_RUN",
    ....
}
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{
    "message":"Validation failed",
    "errors":[
        { "field":"state", "message":"Value is not permitted" }
    ]
}
```

```
{ "message":"Illegal state change (RUNNING->SCHEDULED_TO_RUN)" }
```

```
{ "message":"Illegal state change (RUNNING->STOPPED)" }
```

8.5. Change task working place

```
PATCH /tasks/:id
```

Data parameters

Name	Type	Description
transcodingNodeGroup	number	The ID of nodes group, that will be used for transcoding
transcodingNode	number	The node ID that will be used for transcoding. Required for **SDI** source.
streamingNodeGroup	number	The ID of nodes group, that will be used for streaming
streamingNode	number	The node ID that will be used for streaming

Data example

```
{
    "transcodingNodeGroup": 1,
    "streamingNodeGroup": 1
}
```

8.6. Remove a task

```
DELETE /tasks/:id
```

Success response

```
HTTP/1.1 200 OK
```

Error response

```
HTTP/1.1 400 Bad Request
```

```
{ "message":"Task not found" }
```

```
{ "message":"This action is inappropriate in current task state" }
```

⚠ Task should be stopped before it is deleted

8.7. Run statistics

```
GET /tasks/:id/runstatistics
```

Parameters

Name	Type	Required	Description
from	number	Yes	Start time (milliseconds from the 1970-01-01T00:00:00Z)
to	number	Yes	End time (milliseconds from the 1970-01-01T00:00:00Z)

Example 11. Get statistics

```
curl -s \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-b auth -X GET
[HOST]/api/v1/tasks/1/runstatistics?from=1451606400359&to=1452507496359
```

Success response

```
HTTP/1.1 200 OK
```

```
[
    {
        "dspCores": 7.0,                ①
        "timestamp": 1452507292823,     ②
        "worktime": 9,                  ③
        "taskConfigurationVersion": "1", ④
        "taskConfiguration": { ... },   ⑤
        "codecprofiles": [ { ... }, ... ] ⑥
    },
    {
        "dspCores": 7.0,                ①
        "timestamp": 1452507347830,     ②
        "worktime": 49,                 ③
        "taskConfigurationVersion": "1", ④
        "taskConfiguration": { ... },   ⑤
        "codecprofiles": [ { ... }, ... ] ⑥
    },
    {
        "dspCores": 5.0,                ①
        "timestamp": 1452507419987,     ②
        "worktime": 146,                ③
        "taskConfigurationVersion": "1", ④
        "taskConfiguration": { ... },   ⑤
        "codecprofiles": [ { ... }, ... ] ⑥
    }
]
```

① The count of used cores

② Start time (milliseconds from the 1970-01-01T00:00:00Z)

③ Time of working time (seconds)

④ Version of task configuration and codecprofiles

⑤ Snapshot of task configuration when tasks started

⑥ Codecprofile snapshots (used by the task)

Appendix A: TI H.264 properties

General

entropyCodingMode

Entropy coding type

`0` ⇒ CAVLC (Context-adaptive variable-length coding)

`1` ⇒ CABAC (Context-adaptive binary arithmetic coding)

gopStructure

Type of Group of Pictures (GOP)

`0` ⇒ Open or Non uniform (IBBPBBP)

`1` ⇒ Closed or Uniform (BBIBBPBB)

IDRFrameInterval

Interval b/w two IDR frames

`0` ⇒ IDR BBP I BBP I

`1` ⇒ IDR BBP IDR BBP IDR

`2` ⇒ IDR BBP I BBP IDR

`3` ⇒ IDR BBP I BBP I BBP IDR

transformBlockSize

Transform block size

`0` ⇒ 4x4 only

`1` ⇒ 8x8 only

log2MaxFNumMinus4

Limits the maximum frame number in the bit-stream (1<< (log2MaxFNumMinus4 + 4))

`0` to `12`

picOrderCountType

Picture Order Count (POC) type

`0` ⇒ POC type 0

`1` ⇒ POC type 1

`2` ⇒ POC type 2

interlaceCodingType

Type of interlaced coding

`2` ⇒ MRF (Most Recent Reference Field)

`3` ⇒ ARF (Adaptive Reference Field)

`4` ⇒ SPF (Same Parity Reference Field) (adaptive)

encodingPreset

Encoding preset

`0` ⇒ Default

`1` ⇒ High quality

`2` ⇒ High speed

`3` ⇒ User defined

rateControlPreset

Preset to control rate control selection

`5` ⇒ User defined configuration using extended parameters
`4` ⇒ No configurable video rate control mechanism (None)
`2` ⇒ Variable Bit Rate (VBR)
`1` ⇒ Constant Bit Rate (CBR)

maxInterFrameInterval

This is used for setting the maximum number of B frames between two reference frames. Distance from I-frame to P-frame: 1 - No B-frames 2 - Insert one B-frame. 3 - Insert two B frames N - Insert N-1 B frames between two P frames.

`1` to `255`

minBitRate

Minimum Bit Rate for encoding in bits per second

maxBitRate

Maximum Bit Rate for encoding in bits per second

profile

Encoding profile

`66` ⇒ Baseline Profile
`77` ⇒ Main Profile
`100` ⇒ High Profile

level

Level IDC

`9` ⇒ Level 1b
`10` ⇒ Level 1.0
`11` ⇒ Level 1.1
`12` ⇒ Level 1.2
`13` ⇒ Level 1.3
`20` ⇒ Level 2.0
`21` ⇒ Level 2.1
`22` ⇒ Level 2.2
`30` ⇒ Level 3.0
`31` ⇒ Level 3.1
`32` ⇒ Level 3.2
`40` ⇒ Level 4.0
`41` ⇒ Level 4.1
`42` ⇒ Level 4.2
`50` ⇒ Level 5.0
`51` ⇒ Level 5.1

Rate control

rateControlParamsPreset

Rate control params preset

`0` ⇒ Default
`1` ⇒ User defined

rcAlgo

Rate control algorithm
`0` ⇒ Variable bitrate
`1` ⇒ Constant bitrate (low delay)
`2` ⇒ Constrained CBR (No Frame Skips)

qpI

Initial quantization parameter for I/IDR frames. -1 indicates auto initialization else Initial QP
`-1` to `55`

qpP

Initial quantization parameter for P frames. -1 indicates auto initialization else Initial QP
`-1` to `55`

qpOffsetB

Offset of B frames Quantization Parameter from P frames. qpP + qpOffsetB should be in range of [0,51]
`0` to `51`

qpMaxI

Maximum quantization parameter for I/IDR frame(s)
`0` to `51`

qpMinI

Minimum quantization parameter for I/IDR frame(s)
`0` to `51`

qpMaxP

Maximum quantization parameter for inter frame(s)
`0` to `51`

qpMinP

Minimum quantization parameter for inter frame(s)
`0` to `51`

qpMaxB

Maximum quantization parameter for B frame(s)
`0` to `51`

qpMinB

Minimum quantization parameter for B frame(s)
`0` to `51`

CbQPIndexOffset

Specifies offset to be added to luma QP for addressing QPC values table for chroma component Cb

`-12` to `12`

CrQPIndexOffset

Specifies offset to be added to luma QP for addressing QPC values table for chroma component Cr
`-12` to `12`

initialBufferLevel

Initial buffer level for HRD compliance

HRDBufferSize

Hypothetical reference decoder buffer size. This size controls the frame skip logic of the encoder. For low delay applications this size should be small. This size is in bits.

enablePRC

Control Flag to enable MB level Perceptual Rate Control
`0` ⇒ Disable
`1` ⇒ Enable

frameSkipAfterSceneChange

Frame skip after scene change
`0` ⇒ No forced skip after scene change
`1` ⇒ Force skip frame after coding scene change frame

scalingMatrixPreset

Scaling Matrix selection of encoder available in High Profile only
`0` ⇒ No scaling matrices
`1` ⇒ Normal
`2` ⇒ Noisy
`3` ⇒ Standard default

Inter coding

interCodingPreset

Inter coding mode preset
`0` ⇒ Default values
`1` ⇒ User defined

MvRangeVerP

Vertical motion vector range for P frames in pixels
`16` to `496`

MvRangeHorP

Horizontal motion vector range for P frames
`16` to `496`

MvRangeVerB

Vertical motion vector range for B frames in pixels
`16` to `496`

MvRangeHorB

Horizontal motion vector range for B frames

`16` to `496`

maxMVperMB

Defines the maximum MV per macroblock

`1` ⇒ 1

`4` ⇒ 4

Intra coding

intraCodingPreset

Intra coding mode preset

`0` ⇒ Default values

`1` ⇒ User defined

enableIntraPartition

Enable intra partition

`0` ⇒ NONE

`1` ⇒ ISLICES

`2` ⇒ IPSLICES

`3` ⇒ IBSLICES

`4` ⇒ IPBSLICES

intraRefreshMethod

Rate at which intra MB Refresh is done. e.g. 10 indicates every 10th MB will coded as Intra in inter pictures.

`0` ⇒ NONE

`1` ⇒ CYCLIC_MBS

intraRefreshRate

Rate at which intra MB Refresh is done

constrainedIntraPredEnable

Controls the intra MB coding in inter slices

`0` ⇒ Disable

`1` ⇒ Enable

Slice mode configuration

sliceCodingPreset

Preset value for slice coding mode

`0` ⇒ Default values

`1` ⇒ User defined

streamFormat

Controls the type of stream

`0` ⇒ Byte stream format

`1` ⇒ NALU stream format

sliceMode

Type of slice coding

0 ⇒ Frame based
1 ⇒ Slices are controlled based upon number of Macroblocks
2 ⇒ Slices are controlled based upon number of Bytes (supported for Baseline Profile only)

sliceUnitSize

Number of macroblocks per slice

Loop filter control

loopfilterPreset

Preset value for loop filter operation parameters

0 ⇒ Default values
1 ⇒ User defined

loopfilterDisableIDC

Disable H.264 loop filter

0 ⇒ Filter
1 ⇒ No filter
2 ⇒ No loop filter across slices

filterOffsetA

Alpha offset for loop filter

-12 to 12

filterOffsetB

Beta offset for loop filter

-12 to 12

VUI control params

vuiCodingPreset

Preset value for VUI Control Params

0 ⇒ Default values
1 ⇒ User defined

aspectRatioInfoPresentFlag

Controls the insertion of aspect ratio information in VUI part of bit-stream

0 ⇒ Disable
1 ⇒ Enable

aspectRatioIdc

Aspect ratio IDC

0 ⇒ Unspecified
1 ⇒ 1:1 (square)
2 ⇒ 12:11
3 ⇒ 10:11
4 ⇒ 16:11

5 ⇒ 40:33
6 ⇒ 24:11
7 ⇒ 20:11
8 ⇒ 32:11
9 ⇒ 80:33
10 ⇒ 18:11
11 ⇒ 15:11
12 ⇒ 64:33
13 ⇒ 160:99
14 ⇒ 4:3
15 ⇒ 3:2
16 ⇒ 2:1
255 ⇒ Extended": 255

sampleAspectRatioWidth

Extended aspect ratio width. Use 0 for passthrough.
0 to 65535

sampleAspectRatioHeight

Extended aspect ratio height. Use 0 for passthrough.
0 to 65535

videoSignalTypePresentFlag

Controls the insertion of video signal type in VUI part of bit-stream
0 ⇒ Disable
1 ⇒ Enable

videoFormat

Video signal type
0 ⇒ Component
1 ⇒ PAL
2 ⇒ NTSC
3 ⇒ SECAM
4 ⇒ MAC
5 ⇒ Unspecified

videoFullRangeFlag

Flag to specify range of the pixels
0 ⇒ Off
1 ⇒ On

colourDescriptionPresentFlag

Specifies whether colour primaries, transfer_characteristics and matrix_coefficients are present or not
0 ⇒ Off
1 ⇒ On

colourPrimaries

This controls the colour primaries type in VUI part of bit-stream

`1` ⇒ ITU_R_BT709_5

`2` ⇒ UNSPECIFIED

`4` ⇒ ITU_R_BT470_6_SYSTEM_M

`5` ⇒ ITU_R_BT470_6_SYSTEM_B_G

`6` ⇒ SMPTE_170M

`7` ⇒ SMPTE_240M

`8` ⇒ GENERIC_FILM

transferCharacteristics

Indicates the opto-electronic transfer characteristic of the source picture

`1` ⇒ ITU_R_BT709_5

`2` ⇒ UNSPECIFIED

`4` ⇒ ITU_R_BT470_6_SYSTEM_M

`5` ⇒ ITU_R_BT470_6_SYSTEM_B_G

`6` ⇒ SMPTE_170M

`7` ⇒ SMPTE_240M

`8` ⇒ LINEAR

`9` ⇒ LOG_100_1_RANGE

`10` ⇒ LOG_316_1_RANGE

`11` ⇒ IEC_61966_2_4

`12` ⇒ ITU_R_BT1361

matrixCoefficients

Describes the matrix coefficients used in deriving luma and chroma signals from the green, blue, and red primaries

`0` ⇒ GBR

`1` ⇒ ITU_R_BT709_5

`2` ⇒ UNSPECIFIED

`4` ⇒ USFCC

`5` ⇒ ITU_R_BT470_6_SYSTEM_B_G

`6` ⇒ SMPTE_170M

`7` ⇒ SMPTE_240M

`8` ⇒ YCGCO

timingInfoPresentFlag

Controls the insertion of timing info related parameters in VUI part of bit-stream

`0` ⇒ Off

`1` ⇒ On

Dynamic params

intraFrameInterval

The number of frames between two I frames. 0: IPPPP..., 1: IIII..., 2: IPIPIPIPI, 3: IPPIPPIPPI or IPBIPBIPBI, etc

mvAccuracy

Pixel accuracy of the motion vector

`0` ⇒ Integer pel

2 ⇒ Quarter pel

bitRate

Bit rate

interFrameInterval

Number of B frames between two reference frames; that is, the number of B frames between two P frames or I/P frames. Should be less than maxInterFrameInterval. intraFrameInterval should be multiple of interFrameInterval

1 to 255

A.1. Sample

Default values

```
{
  "CbQPIndexOffset": 0,
  "CrQPIndexOffset": 0,
  "HRDBufferSize": 8000000,
  "IDRFrameInterval": 1,
  "MvRangeHorB": 144,
  "MvRangeHorP": 144,
  "MvRangeVerB": 32,
  "MvRangeVerP": 32,
  "aspectRatioIdc": 255,
  "aspectRatioInfoPresentFlag": 1,
  "bitRate": 8000000,
  "colourDescriptionPresentFlag": 1,
  "colourPrimaries": 5,
  "constrainedIntraPredEnable": 0,
  "enableIntraPartition": 1,
  "enablePRC": 1,
  "encodingPreset": 3,
  "entropyCodingMode": 0,
  "filterOffsetA": 0,
  "filterOffsetB": 0,
  "frameSkipAfterSceneChange": 1,
  "gopStructure": 0,
  "initialBufferLevel": 8000000,
  "interCodingPreset": 1,
  "interFrameInterval": 1,
  "interlaceCodingType": 4,
  "intraCodingPreset": 1,
  "intraFrameInterval": 24,
  "intraRefreshMethod": 1,
  "intraRefreshRate": 0,
  "level": 40,
  "log2MaxFNumMinus4": 10,
  "loopfilterDisableIDC": 2,
  "loopfilterPreset": 1,
  "matrixCoefficients": 5,
  "maxBitRate": 16000000,
  "maxInterFrameInterval": 1,
  "maxMVperMB": 1,
  "minBitRate": 8000000,
  "mvAccuracy": 0,
  "picOrderCountType": 0,
  "profile": 100,
  "qpI": -1,
  "qpMaxB": 35,
  "qpMaxI": 35,
  "qpMaxP": 35,
  "qpMinB": 10,
  "qpMinI": 10,
  "qpMinP": 10,
  "qpOffsetB": 4,
  "qpP": -1,
  "rateControlParamsPreset": 1,
  "rateControlPreset": 5,
  "rcAlgo": 0,
  "sampleAspectRatioHeight": 0,
  "sampleAspectRatioWidth": 0,
  "scalingMatrixPreset": 1,
  "sliceCodingPreset": 0,
  "sliceMode": 0,
  "sliceUnitSize": 0,
  "streamFormat": 0,
  "timingInfoPresentFlag": 1,
  "transferCharacteristics": 5,
  "transformBlockSize": 0,
  "videoFormat": 2,
  "videoFullRangeFlag": 0,
  "videoSignalTypePresentFlag": 0,
  "vuiCodingPreset": 1
}
```

Appendix B: AAC properties

sample_rate

Samples per second

bit_rate

The average bitrate

channels

Number of audio channels

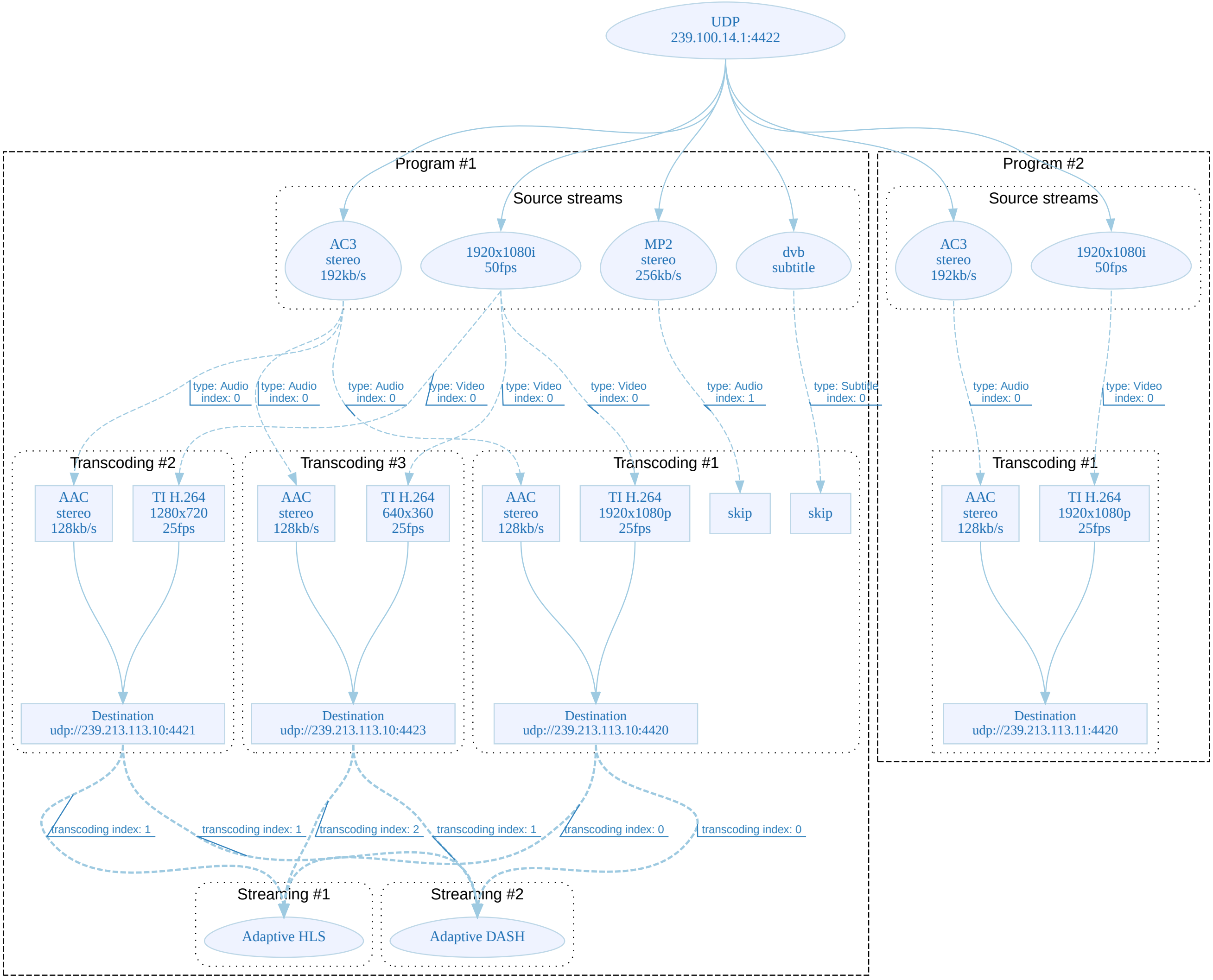
B.1. Sample

```
{
  "bit_rate": 64000,
  "sample_rate":48000,
  "channels":2
}
```

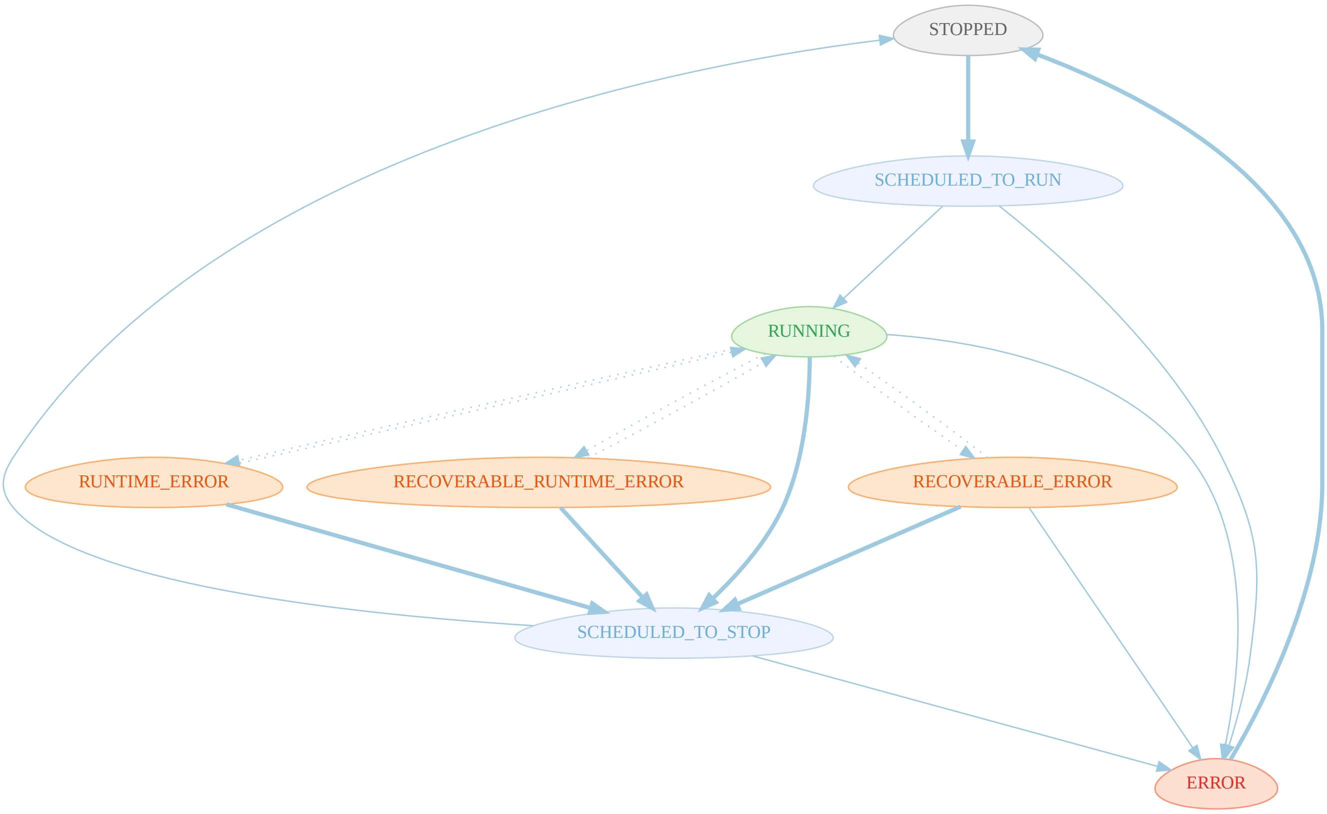
Appendix C: Deltacast SDI standards

Id	Name	Description
0	SMPTE 274M	HD 1080p @ 25Hz standard
1	SMPTE 274M	HD 1080p @ 30Hz standard
2	SMPTE 274M	HD 1080i @ 50Hz standard
3	SMPTE 274M	HD 1080i @ 60Hz standard
4	SMPTE 296M	HD 720p @ 50Hz standard
5	SMPTE 296M	HD 720p @ 60Hz standard
6	SMPTE 259M	SD PAL standard
7	SMPTE 259M	SD NTSC standard
8	SMPTE 274M	HD 1080p @ 24Hz standard
9	SMPTE 274M	3G 1080p @ 60Hz standard
10	SMPTE 274M	3G 1080p @ 50Hz standard
11	SMPTE 274M	HD 1080psf @ 24Hz standard
12	SMPTE 274M	HD 1080psf @ 25Hz standard
13	SMPTE 274M	HD 1080psf @ 30Hz standard
14	SMPTE 296M	HD 720p @ 24Hz standard
15	SMPTE 296M	HD 720p @ 25Hz standard
16	SMPTE 296M	HD 720p @ 30Hz standard
17	SMPTE 2048M	HD 2048p @ 24 Hz standard
18	SMPTE 2048M	HD 2048p @ 25 Hz standard
19	SMPTE 2048M	HD 2048p @ 30 Hz standard
20	SMPTE 2048M	HD 2048psf @ 24 Hz standard
21	SMPTE 2048M	HD 2048psf @ 25 Hz standard
22	SMPTE 2048M	HD 2048psf @ 30 Hz standard
23	SMPTE 2048M	3G 2048p @ 60Hz standard
24	SMPTE 2048M	3G 2048p @ 50Hz standard
25	SMPTE 2048M	3G 2048p @ 48Hz standard

Appendix D: Transcoding schema



Appendix E: Task state diagram



Appendix F: SDI task

Listing 5. Data example

```
{
    "name": "Some SDI task",
    "description": "Descriprion for SDI task",
    "sourceType": "SDI",
    "state": "STOPPED",
    "transcodingNode": 1,
    "transcodingNodeGroup": 1,
    "streamingNode": null,
    "streamingNodeGroup": 2,
    "sourceParameters": {
        "type": "SDI",
        "boardId": 0,
        "inputId": 0
    },
    "programs": [
        {
            "pid": 1001,
            "serviceName":
            "some service",
            "streams": [
                {
                    "type": "Video",
                    "index": 0,
                    "mediaAttributes": {
                        "type": "Video",
                        "frameRate": 25,
                        "maxWidth":  1920,
                        "maxHeight": 1080,
                        "frameScanType": "Progressive"
                    }
                },
                {
                    "type": "Audio",
                    "index": 0,
                    "mediaAttributes": { "type": "Audio" },
                    "channels": [0, 1]
                },
                {
                    "type": "Audio",
                    "index": 1,
                    "mediaAttributes": { "type": "Audio" },
                    "channels": [2, 3, 4, 5, 6]
                }
            ],
            "transcodings": [
                {
                    "destination": "udp://239.213.113.10:4420",
                    "streams": [
                        {
                            "index": 0,
                            "type": "Video",
                            "codecProfile": 1
                        },
                        {
                            "index": 0,
                            "type": "Audio",
                            "codecProfile": 2
                        },
                        {
                            "index": 1,
                            "type": "Audio",
                            "codecProfile": 3
                        }
                    ]
                },
                {
                    "destination": "udp://239.213.113.10:4421",
                    "streams": [
                        {
                            "index": 0,
                            "type": "Video",
                            "codecProfile": 1,
                            "mediaAttributes": {
                                "type": "Video",
                                "maxWidth": 960,
                                "maxHeight": 540
                            }
                        },
                        {
                            "index": 0,
                            "type": "Audio",
                            "codecProfile": 2
                        },
                        {
                            "index": 1,
                            "type": "Audio",
                            "codecProfile": 3
                        }
                    ]
                },
                {
                    "destination": "udp://239.213.113.10:4422",
                    "streams": [
                        {
                            "index": 0,
                            "type": "Video",
                            "codecProfile": 1,
                            "mediaAttributes": {
                                "type": "Video",
                                "maxWidth": 480,
                                "maxHeight": 270
                            }
                        },
                        {
                            "index": 0,
                            "type": "Audio",
                            "codecProfile": 2
                        },
                        {
                            "index": 1,
                            "type": "Audio",
                            "codecProfile": 3
                        }
                    ]
                }
            ],
            "streamings": [
                {
                    "type": "HLS",
                    "transcodingIndexes": [ 0, 1, 2 ]
                }
            ]
        }
    ]
}
```

Appendix G: UDP task

Listing 6. Data example

```
{
    "name": "Some UDP task",
    "description": "Descriprion for UDP task",
    "sourceType": "UDP",
    "state": "STOPPED",
    "transcodingNode": null,
    "transcodingNodeGroup": 1,
    "streamingNode": null,
    "streamingNodeGroup": 2,
    "sourceParameters": {
        "type": "UDP",
        "source": "udp://239.100.7.1:4422"
    },
    "programs": [
        {
            "pid": 28107,
            "serviceName": "Bayerisches",
            "serviceProvider": "ARD",
            "transcodingHints": { "decoding": "HOST" },
            "streams": [
                {
                    "type": "Video",
                    "index": 0,
                    "mediaAttributes": {
                        "type": "Video",
                        "maxWidth": 720,
                        "maxHeight": 576,
                        "frameScanType": "Dynamic",
                        "frameRate": 25
                    }
                },
                {
                    "type": "Audio",
                    "index": 0,
                    "mediaAttributes": { "type": "Audio" }
                },
                {
                    "type": "Audio",
                    "index": 1,
                    "mediaAttributes": { "type": "Audio" }
                },
                {
                    "type": "Audio",
                    "index": 2,
                    "mediaAttributes": { "type": "Audio" }
                },
                {
                    "type": "Subtitle",
                    "index": 0,
                    "mediaAttributes": { "type": "Subtitle" }

                },
                {
                    "type": "Subtitle",
                    "index": 1,
                    "mediaAttributes": { "type": "Subtitle" }
                }
            ],
            "transcodings": [{
                "destination": "udp://239.100.100.1:4000",
                "streams": [{
                    "type": "Video",
                    "index": 0,
                    "mediaAttributes": {
                        "type": "Video",
                        "maxWidth": null,
                        "maxHeight": null,
                        "frameScanType": "Progressive",
                        "frameRate": null
                    },
                    "codecProfile": 1
                },
                {
                    "type": "Audio",
                    "index": 0,
                    "mediaAttributes": { "type": "Audio" },
                    "codecProfile": 2
                },
                {
                    "type": "Audio",
                    "index": 1,
                    "mediaAttributes": { "type": "Audio" },
                    "codecProfile": null,
                    "skip": true
                },
                {
                    "type": "Audio",
                    "index": 2,
                    "mediaAttributes": { "type": "Audio" },
                    "codecProfile": null,
                    "skip": true
                },
                {
                    "type": "Subtitle",
                    "index": 0,
                    "mediaAttributes": { "type": "Subtitle" },
                    "skip": true
                },
                {
                    "type": "Subtitle",
                    "index": 1,
                    "mediaAttributes": { "type": "Subtitle" },
                    "skip": true
                }]
            },
            {
                "destination": "udp://239.100.100.1:4001",
                "streams": [{
                    "type": "Video",
                    "index": 0,
                    "mediaAttributes": {
                        "type": "Video",
                        "maxWidth": 600,
                        "maxHeight": 480,
                        "frameScanType": "Progressive",
                        "frameRate": null
                    },
                    "codecProfile": 1
                },
                {
                    "type": "Audio",
                    "index": 0,
                    "mediaAttributes": { "type": "Audio" },
                    "codecProfile": 2
                }]
            },
            {
                "destination": "udp://239.100.100.1:4002",
                "streams": [{
                    "type": "Video",
                    "index": 0,
                    "mediaAttributes": {
                        "type": "Video",
                        "maxWidth": 480,
                        "maxHeight": 384,
                        "frameScanType": "Progressive",
                        "frameRate": null
                    },
                    "codecProfile": 1
                },
                {
                    "type": "Audio",
                    "index": 0,
                    "mediaAttributes": { "type": "Audio" },
                    "codecProfile": 2
                }]
            }],
            "streamings": [
```

```json
                {
                    "type": "HLS",
                    "transcodingIndexes": [0,1,2]
                }
            ]
        }
    ]
}
```