Elecard Converter Studio VOD v.1.2

User Guide

User Guide Notices

Elecard Converter Studio VOD v.1.2 User Guide
First edition: September, 2017.
Date modified: July 29, 2022.

For information, contact Elecard.
More information can be found at: [https://www.elecard.com.](https://www.elecard.com)
For Technical Support, please contact the Elecard Technical Support Team: tsup@elecard.com

1 Introduction

1.1 Preface

Elecard Converter Studio VOD is a professional solution for encoding and transcoding multimedia data into highly demanded video formats such as MPEG-2/AVC/HEVC with up to 16K resolution, supporting adaptive streaming technologies over HLS/MPEG-DASH. Operating on a hardware that has passed through comprehensive testing, Elecard Converter Studio VOD provides high performance and continuous content delivery suitable for projects of any scale and complexity.

1.2 Using This Guide

This Guide allows a user finding information on basic functions and operation principles of Elecard Converter Studio VOD software solution. The Guide describes installation and configuration rules for Elecard Converter Studio VOD.

1.3 System Requirements

System requirements depend on several factors:

- type of task (encoding, transcoding etc.);
- input and output formats, bitrate and resolution;
- signal preprocessing (upscaling, downscaling, deinterlacing, etc.);
- the number of channels to be processed.

The minimal system requirements recommended for proper operation of Elecard Converter Studio VOD:

CPU encoding	**GPU encoding (Intel® QuickSync)**
• CPU (Intel® 4/5/6/7 generation, Xeon) • 8 GB RAM (two-channels mode is required) • Any Graphic card • Windows® 8/10/server 2012/2016 (64 bit)	• CPU (Intel® 5/6/7 generation, Xeon) • 8 GB RAM (two-channels mode is required) • Intel Graphics • Windows® 8/10 (64 bit)

1.4 Activating Converter Studio VOD

There are two ways to activate Elecard Converter Studio VOD as described below:

- HWkey License.

Unique identifying code is compiled based on target PC hardware configuration with the help of a special utility, and Converter Studio VOD build is tied to this code. To receive the utility, contact Elecard Technical Support Team or your project coordinator.

- HASP License.

Elecard mails a HASP key which has a unique code and should be plugged into a PC where Converter Studio VOD is going to be used.

1.5 Licensing and Technical Support

Elecard Converter Studio VOD is available as a demo version or a registered product.

By installing, copying, or otherwise using the SOFTWARE PRODUCT or any UPDATES, you agree to be bound by the terms of the "Elecard" End-User License Agreement ("EULA"). This EULA is a legal agreement between you (either an individual or a single entity) and Elecard for the "Elecard" software product(s) accompanying this EULA, which include(s) computer software and may include "online" or electronic documentation, associated media, and printed materials ("SOFTWARE PRODUCT").

For sales and licensing information, contact the Elecard Sales Department: sales@elecard.com.

For technical support, contact Elecard Technical Support Team at: tsup@elecard.com.

2 Describing Converter Studio VOD

2.1 Installing Converter Studio VOD

1. Run Elecard Converter Studio VOD Setup.
2. Follow the on-screen instructions to complete the installation.
3. When setup has finished installing all the necessary files on your computer, the *Elecard Converter Studio VOD has been successfully installed* dialog box will appear, and the program is ready to run. You do not need to reboot your computer.

Before Elecard Converter Studio VOD is installed, take into consideration the following aspects:

- CPU performance is of fundamental importance for real time encoding process. Take it into account when choosing hardware for the encoding server;
- the disk selected for installation should have enough free space for saving the system log (for highly detailed logs) and for memory dump files (in case of the application abnormal termination);
- if the QuickSync hardware acceleration is used for encoding, an Intel Graphics card is required;
- to control the encoding server over SNMP, the corresponding service should be installed in the system.

2.2 Uninstalling Converter Studio VOD

To uninstall Elecard Converter Studio VOD:

1. Open the folder C:\Program Files\Elecard\Elecard Converter Studio VOD and select Uninstall.
2. Follow the on-screen instructions to complete Elecard Converter Studio VOD uninstallation.

2.3 Running Converter Studio VOD

To run Elecard Converter Studio VOD, follow the steps below:

1. start the browser (Google Chrome is preferred);
2. type **localhost:8888** in the address bar;
3. if Converter Studio VOD is installed on another hardware, type **ip:8888** in the address bar;
4. The Converter Studio VOD control panel should open.

2.4 Technical Requirements

Elecard Converter Studio VOD supports the following formats and protocols:

Input formats

Interfaces:

- HTTP
- FTP
- File
- Watch folder

Containers:

- AVI
- MP4
- TS
- MOV
- MXF
- MPG
- MKV

- M2TS

Video codecs:

- HEVC/H.265
- AVC/H.264
- MPEG-2
- MPEG-1

Audio codecs:

- AAC/HE-AAC v1
- MPEG-1/2 Layer I/II/III
- AC3
- PCM
- Any

Output formats

Containers and protocols:

- TS
- MP4
- HLS
- MPEG-DASH

Video codecs:

- HEVC/H.265
- AVC/H.264
- MPEG-2

Audio codecs:

- AAC/HE-AAC v1

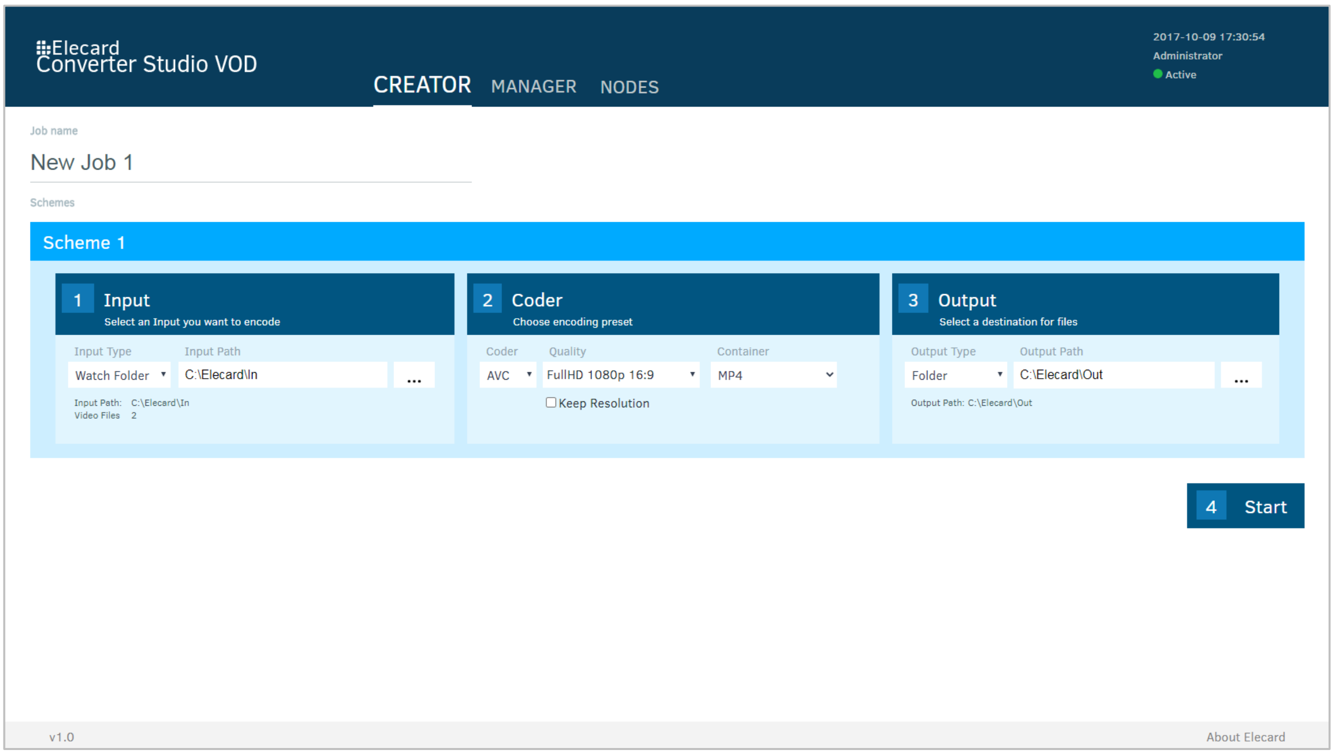
2.5 Features

Elecard Converter Studio VOD implements the following functionality:

- User-friendly GUI for creating transcoding schemes;
- Receiving files over HTTP;
- Receiving files over FTP;
- Watch folder: automatic encoding of new files that appear in the specified folder;
- Encoding into AVC, HEVC, MPEG-2 formats
- HLS and DASH protocols support;
- API for integration of Converter Studio VOD into existing workflow;
- Hardware acceleration via Intel Quick Sync.

2.6 Converter Studio VOD WEB Manager

The application main window has an information box located in the upper right corner and functional tabs located at the top of the window (the Creator, Manager and Nodes tabs).



2017-10-09 17:30:54
Administrator
● Active

Job name
New Job 1

Schemes

Scheme 1

1 Input
Select an Input you want to encode

Input Type	Input Path
Watch Folder	C:\Elecard\In

Input Path: C:\Elecard\In
Video Files 2

2 Coder
Choose encoding preset

Coder	Quality	Container
AVC	FullHD 1080p 16:9	MP4

☐ Keep Resolution

3 Output
Select a destination for files

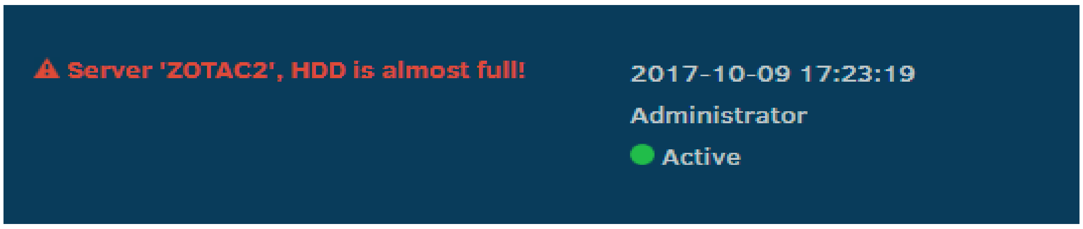
Output Type	Output Path
Folder	C:\Elecard\Out

Output Path: C:\Elecard\Out

Figure 1. Main Window - Converter Studio VOD WEB Manager

2.6.1 Information Box

Information box displays time and server state along with notifications (e.g. low disk space).



⚠ Server 'ZOTAC2', HDD is almost full!

2017-10-09 17:23:19
Administrator
● Active

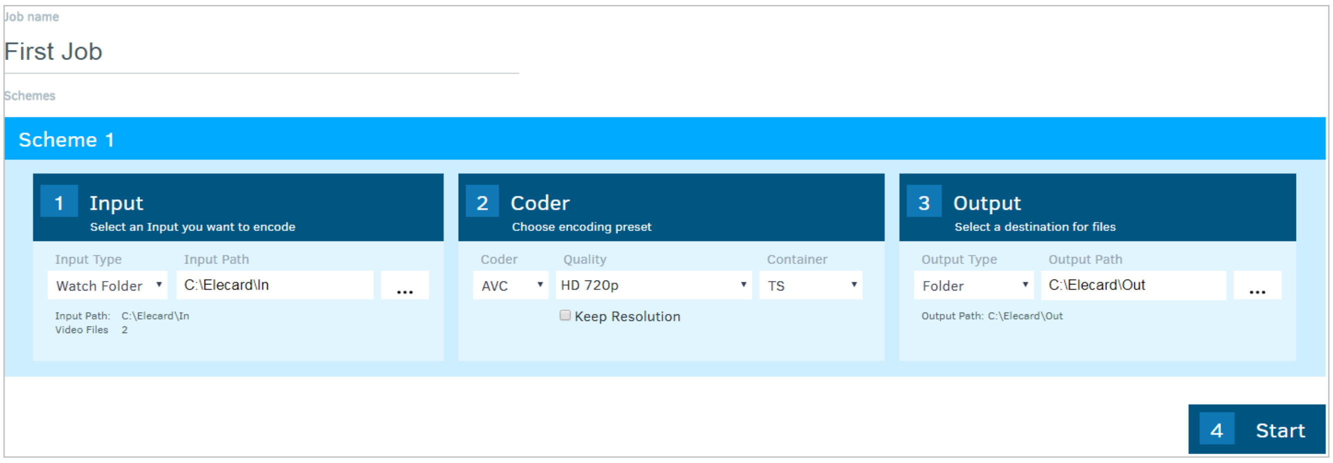
Figure 2. Information box

2.6.2 CREATOR Tab

A transcoding scheme is created in the CREATOR tab. The tab consists of several parts:

- The Job name field, where a task name is set;
- Input section that allows selecting one of four sources:
  - folder – all media files contained in the selected folder will be processed; new files added to the folder after the scheme creation will be processed as well;
  - file – just the selected file will be processed;
  - http – the file will be downloaded from the specified address and processed;
  - ftp – the file will be downloaded from the specified address and processed.
- The Coder section that allows configuring a transcoding scheme:
  - Coder – video compression format,
  - Quality – video resolution for output files; allows selecting one or several presets;
  - Container/Protocol – container or protocol used for an output video stream;
  - Keep Resolution checkbox – allows setting resolution for the output video file that is equal to the input video file resolution.
- Output section that specifies a folder for saving output files;

- Start button that creates the scheme and starts transcoding.


Job name: First Job

Schemes: Scheme 1

1 Input — Select an Input you want to encode
Input Type: Watch Folder
Input Path: C:\Elecard\In
Input Path: C:\Elecard\In
Video Files 2

2 Coder — Choose encoding preset
Coder: AVC
Quality: HD 720p
Container: TS
Keep Resolution

3 Output — Select a destination for files
Output Type: Folder
Output Path: C:\Elecard\Out
Output Path: C:\Elecard\Out

4 Start

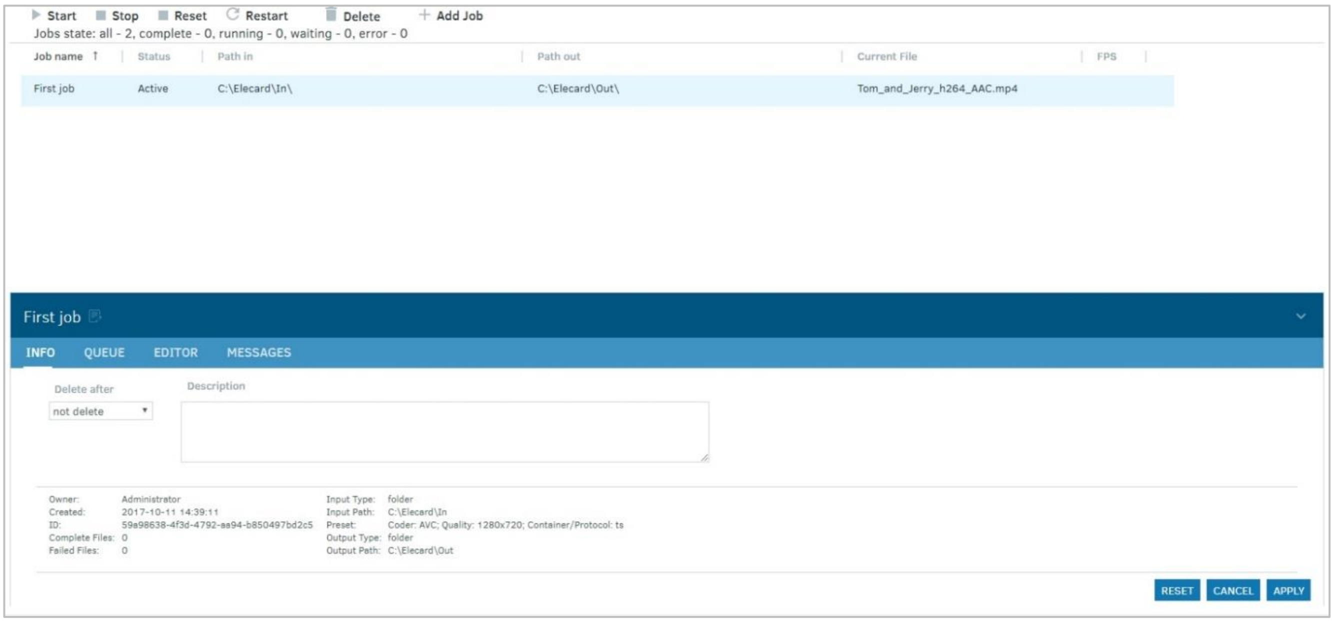
Figure 3. CREATOR tab

2.6.3 MANAGER Tab

The tasks can be monitored, managed, and edited in the MANAGER tab. The tab includes three components:

- The Control panel with buttons at the top of the window where you can start, stop, reset, restart, delete the task or add a new one;
- A list of jobs in the middle of the window where the job name, status, in and out paths, current file and FPS (frames per second) are displayed;
- The Task monitoring and editing panel at the bottom of the window.

You can reorganize the created tasks by clicking the required column name. Up or down arrow should appear.



Jobs state: all - 2, complete - 0, running - 0, waiting - 0, error - 0

Job name ↑	Status	Path in	Path out	Current File	FPS
First job	Active	C:\Elecard\In\	C:\Elecard\Out\	Tom_and_Jerry_h264_AAC.mp4	

First job

Delete after: not delete
Description:

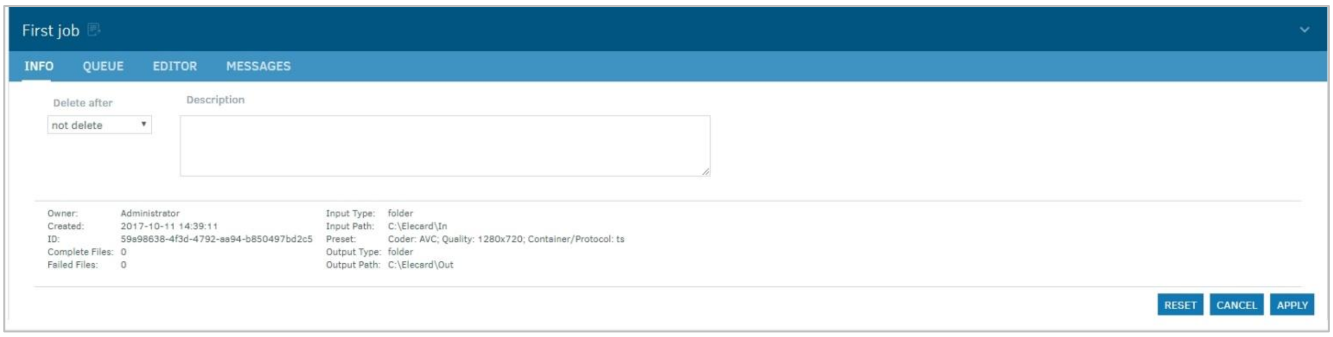
Owner:	Administrator	Input Type:	folder
Created:	2017-10-11 14:39:11	Input Path:	C:\Elecard\In
ID:	59a98638-4f3d-4792-aa94-b850497bd2c5	Preset:	Coder: AVC; Quality: 1280x720; Container/Protocol: ts
Complete Files:	0	Output Type:	folder
Failed Files:	0	Output Path:	C:\Elecard\Out

Figure 4. MANAGER Tab

The monitoring and editing panel includes four tabs INFO, QUEUE, EDITOR and MESSAGES.

2.6.3.1 INFO Tab

The INFO tab includes task information, task description, and time remaining to task auto deletion upon transcoding completion.



Delete after: not delete

Description:

Owner:	Administrator	Input Type:	folder
Created:	2017-10-11 14:39:11	Input Path:	C:\Elecard\In
ID:	59a98638-4f3d-4792-aa94-b850497bd2c5	Preset:	Coder: AVC; Quality: 1280x720; Container/Protocol: ts
Complete Files:	0	Output Type:	folder
Failed Files:	0	Output Path:	C:\Elecard\Out

Figure 5. INFO Tab

Task information shows task owner name, date of task creation, task ID, file status, input and output source types, input and output paths, chosen presets.

The description field is optional.

Task auto deletion time upon completion is set by a user by selecting the preferred period from the drop-down box.

To restore previous settings and comments in the Info tab, click the Reset button.

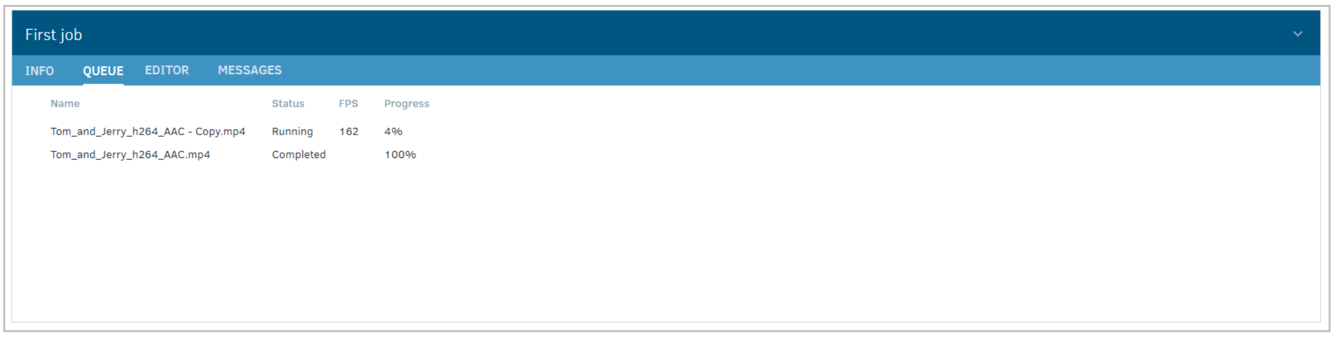
To reject the settings and comments inserted in the Info tab, click the Cancel button.

To save the settings and comments in the Info tab, click the Apply button.

2.6.3.2 QUEUE Tab

The QUEUE tab contains a list of processed files, their status, frames per second and progress of a task processing in percentage. The status can be as follows:

- Configuring – file preparation for the processing queue,
- Running – a file is being processed,
- Success – a file is successfully processed,
- Waiting – a file is in queue for further processing,
- Fault – a processing error,
- Critical Error – a processing fatal error.



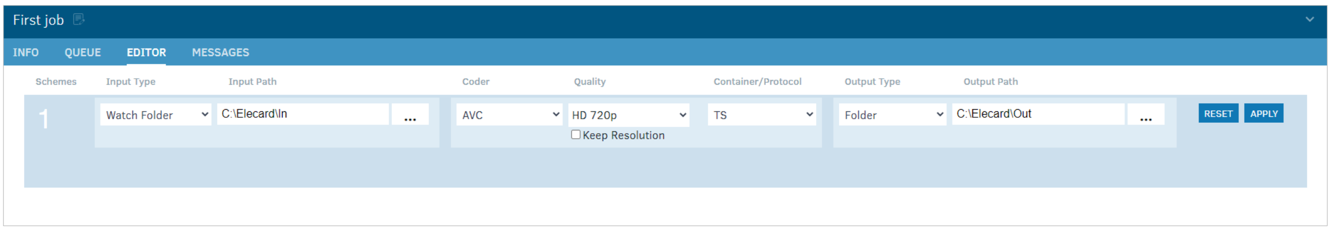
Name	Status	FPS	Progress
Tom_and_Jerry_h264_AAC - Copy.mp4	Running	162	4%
Tom_and_Jerry_h264_AAC.mp4	Completed		100%

Figure 6. QUEUE Tab

2.6.3.3 EDITOR Tab

A task can be edited via the EDITOR tab. The Editor panel looks like the Creator tab panel. To edit the created task, adjust the data and click Apply.

To reset all the settings inserted in the Editor tab to the previous settings, click the Reset button.



First job

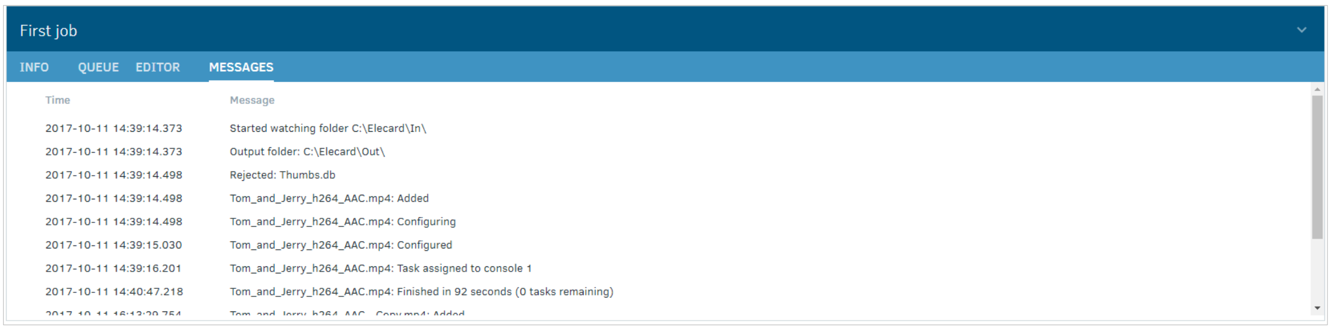
INFO QUEUE EDITOR MESSAGES

Schemes	Input Type	Input Path	Coder	Quality	Container/Protocol	Output Type	Output Path	
1	Watch Folder	C:\Elecard\In ...	AVC	HD 720p ☐ Keep Resolution	TS	Folder	C:\Elecard\Out ...	RESET APPLY

Figure 7. EDITOR Tab

2.6.3.4 MESSAGES Tab

The MESSAGES tab contains a task log, where all actions and their execution time are enlisted.



First job

INFO QUEUE EDITOR MESSAGES

```
Time                      Message
2017-10-11 14:39:14.373   Started watching folder C:\Elecard\In\
2017-10-11 14:39:14.373   Output folder: C:\Elecard\Out\
2017-10-11 14:39:14.498   Rejected: Thumbs.db
2017-10-11 14:39:14.498   Tom_and_Jerry_h264_AAC.mp4: Added
2017-10-11 14:39:14.498   Tom_and_Jerry_h264_AAC.mp4: Configuring
2017-10-11 14:39:15.030   Tom_and_Jerry_h264_AAC.mp4: Configured
2017-10-11 14:39:16.201   Tom_and_Jerry_h264_AAC.mp4: Task assigned to console 1
2017-10-11 14:40:47.218   Tom_and_Jerry_h264_AAC.mp4: Finished in 92 seconds (0 tasks remaining)
```

Figure 8. MESSAGES Tab

2.6.4 NODES Tab

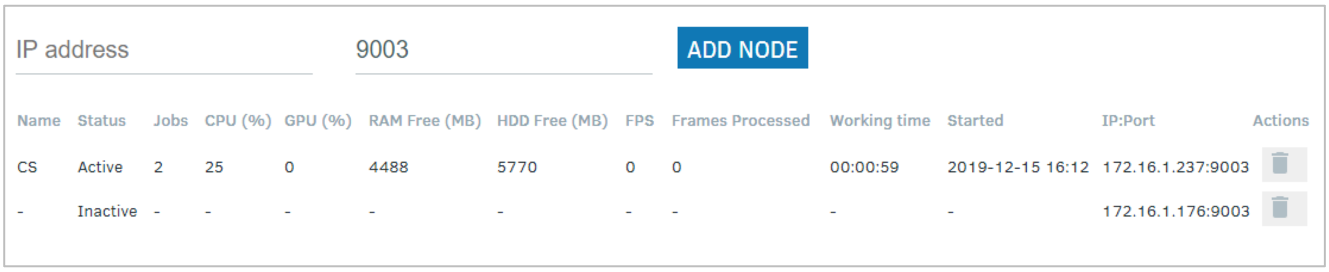
The NODES tab allows to configure monitoring and manage the encoding nodes.

The tab includes two sections:

- Elements for adding an encoding node which are located at the top of the window.
- List of encoding nodes at the bottom of the window.

The list of encoding nodes contains the following information:

- Name – specifies the server name;
- Status – specifies the server status:
  - Active – indicates the server proper operation;
  - Inactive – indicates that the server is not running.
- Jobs – displays a list of active tasks;
- CPU – specifies CPU load in percentage;
- GPU – specifies GPU load in percentage;
- RAM free – displays RAM free space, in Mb;
- HDD free - displays HDD free space, in Mb;
- FPS – speed of frame processing (frames per second);
- Processed Frames – shows the number of processsed frames;
- Working time – displays time passed since the start of the application;
- Started – displays precise time of the application start;
- IP:Port – displays the server IP address and port;
- Actions – the encoding server managing elements (in the current implemetation, removing the node is only supported).



IP address | 9003 | ADD NODE

Name	Status	Jobs	CPU (%)	GPU (%)	RAM Free (MB)	HDD Free (MB)	FPS	Frames Processed	Working time	Started	IP:Port	Actions
CS	Active	2	25	0	4488	5770	0	0	00:00:59	2019-12-15 16:12	172.16.1.237:9003	
-	Inactive	-	-	-	-	-	-	-	-	-	172.16.1.176:9003	

Figure 9. ADD NODE Tab

2.7 Mounting Network Drives

To select files and folders from network-attached storage in the WEB manager, follow the steps below:

- create a text file (txt) with a file name containing Latin characters on a local disk of the hardware where Converter Studio VOD has been installed;
- create a string in the created text file as follows:

net use <DISKNAME>: <PATH> <PASSWORD> /user:<SERVERADRESS>\<USERNAME> /persistent:yes

where:

- <DISKNAME> - the drive letter assigned to the network folder being connected (the specified letter should not be used by other drives);
- <PATH> - a path to the network folder;
- <PASSWORD> - your account password;
- <SERVERADRESS> - an address of a network-attached storage;
- <USERNAME> - your account login.

Example:

net use L: \\192.168.11.11\public 123456 /user:192.168.18.18\IvanIvanov /persistent:yes

- change the file extension from txt to bat;
- open the Task Scheduler (Start –> Control Panel –> Administration);
- click «Create a task…»;

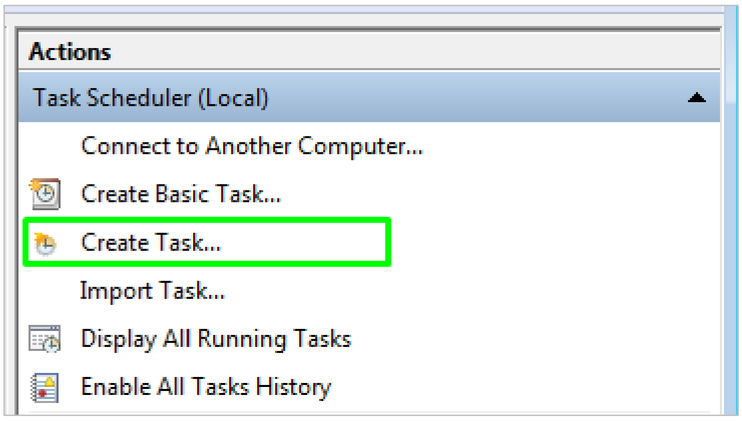
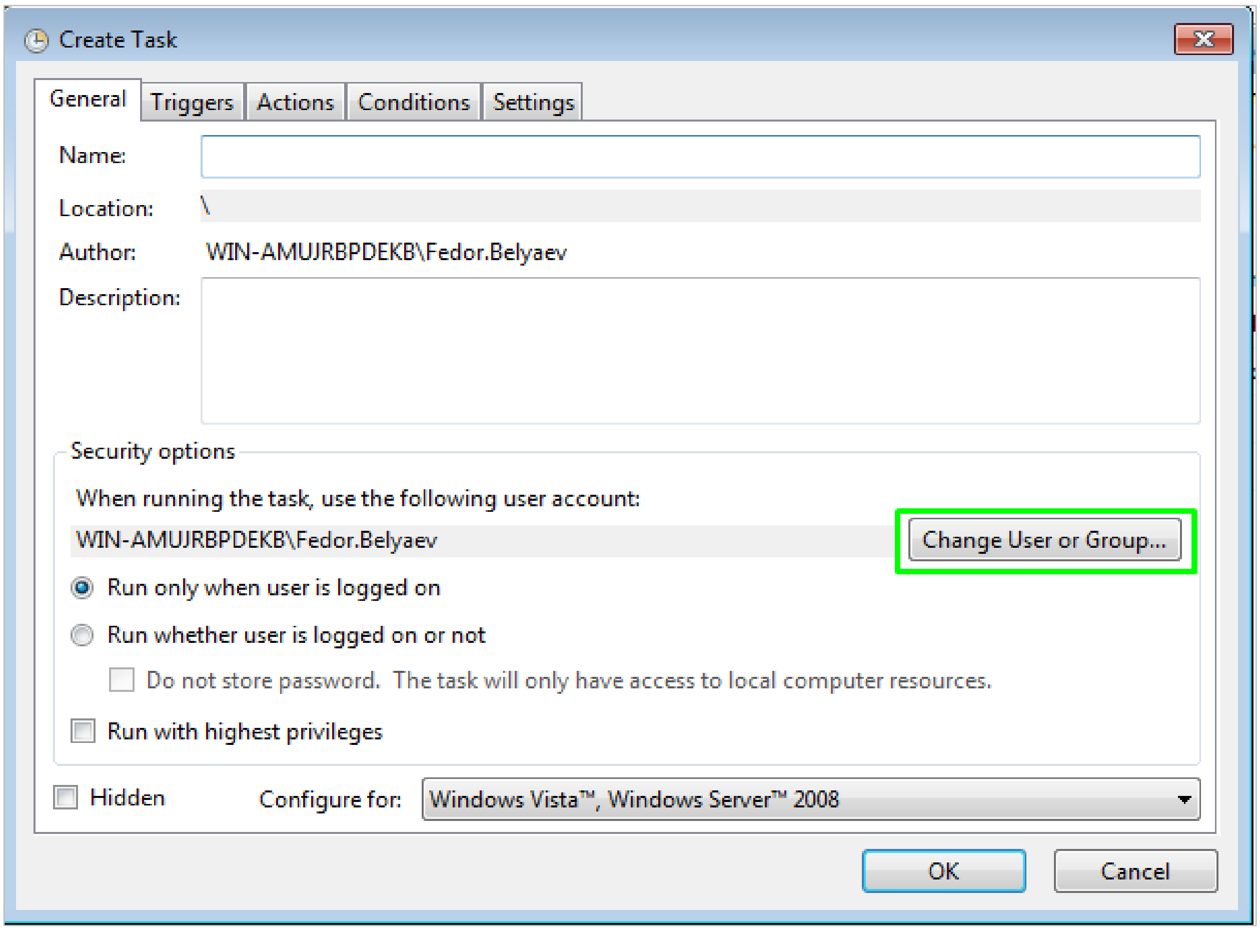


Figure 10. Creating a Task

- specify a task name in the field «Name»;
- select «edit…» to edit a user account;



Create Task

General | Triggers | Actions | Conditions | Settings

Name:

Location: \

Author: WIN-AMUJRBPDEKB\Fedor.Belyaev

Description:

Security options

When running the task, use the following user account:

WIN-AMUJRBPDEKB\Fedor.Belyaev

Change User or Group...

[✓ Run only when user is logged on]

Run whether user is logged on or not

☐ Do not store password. The task will only have access to local computer resources.

☐ Run with highest privileges

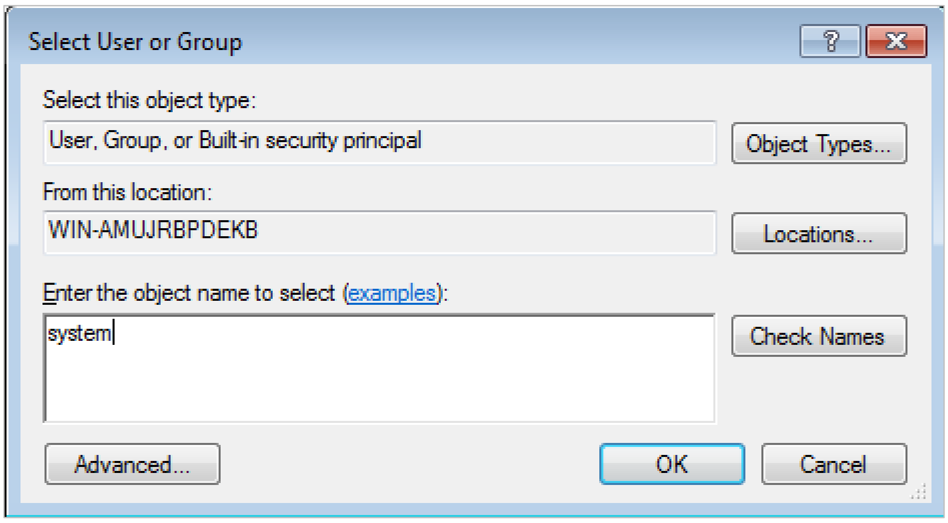
☐ Hidden

Configure for: Windows Vista™, Windows Server™ 2008

OK | Cancel

Figure 11. Creating a task – Editing an Account

- depending on the OS language, enter SYSTEM in the field and click OK;



Select User or Group

Select this object type:
User, Group, or Built-in security principal

Object Types...

From this location:
WIN-AMUJRBPDEKB

Locations...

Enter the object name to select (examples):
system

Check Names

Advanced... | OK | Cancel

Figure 12. Creating a task – Selecting an Object Type

- set a checkbox “Run with highest privileges”;

☑ Run with highest privileges

- open the Triggers tab;
- click the Create button;

- open the drop-down box “Begin the task” and select At startup;

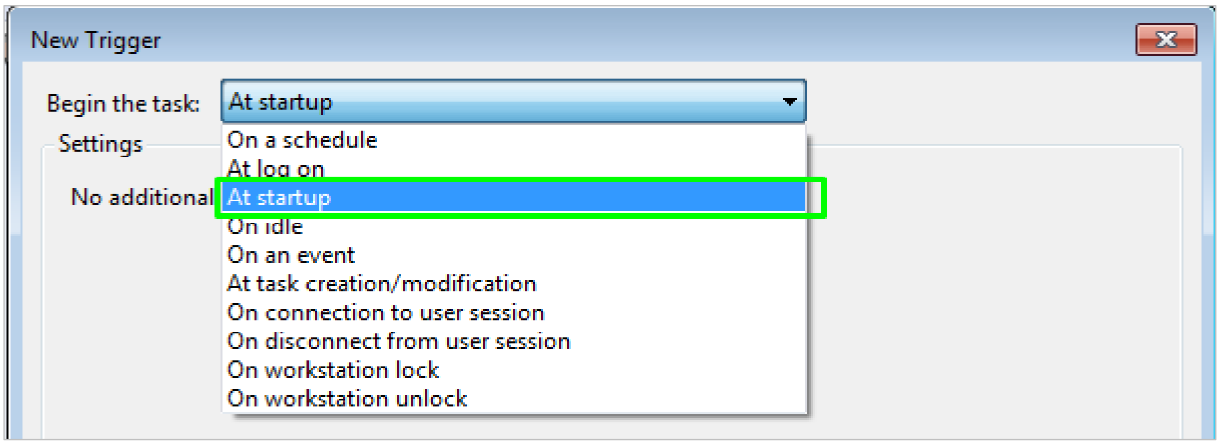


Figure 13. Creating a Trigger – Task startup options

- go to the Actions tab;
- click the Create button;
- select Run startup from the Actions drop-down menu;
- click the Browse button and select the created bat file;
- do not create the Conditions and Parameters tabs;
- create a task by clicking OK;
- go to the Task Scheduler Library folder, right-click the task and select Run. Reboot is not required;

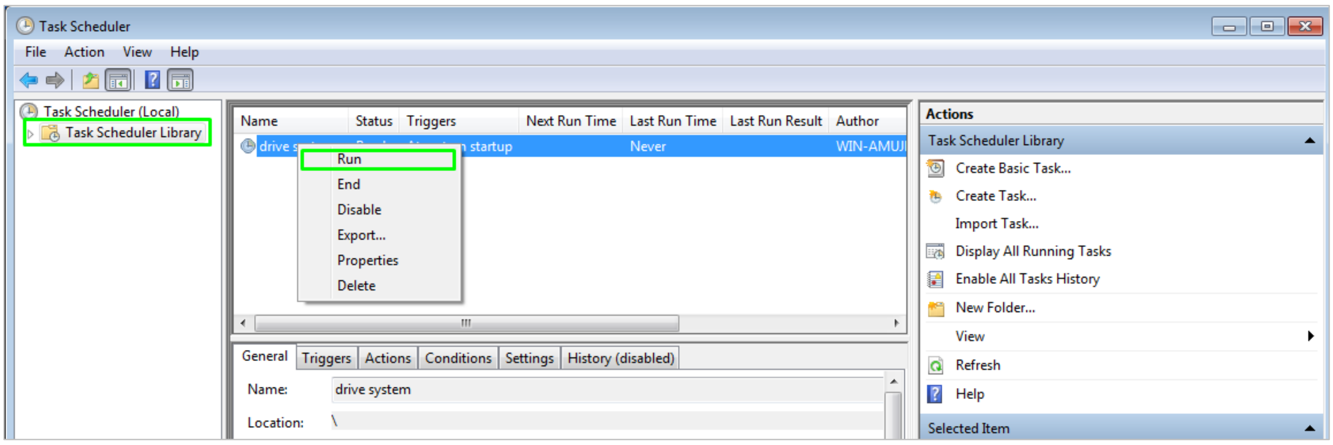
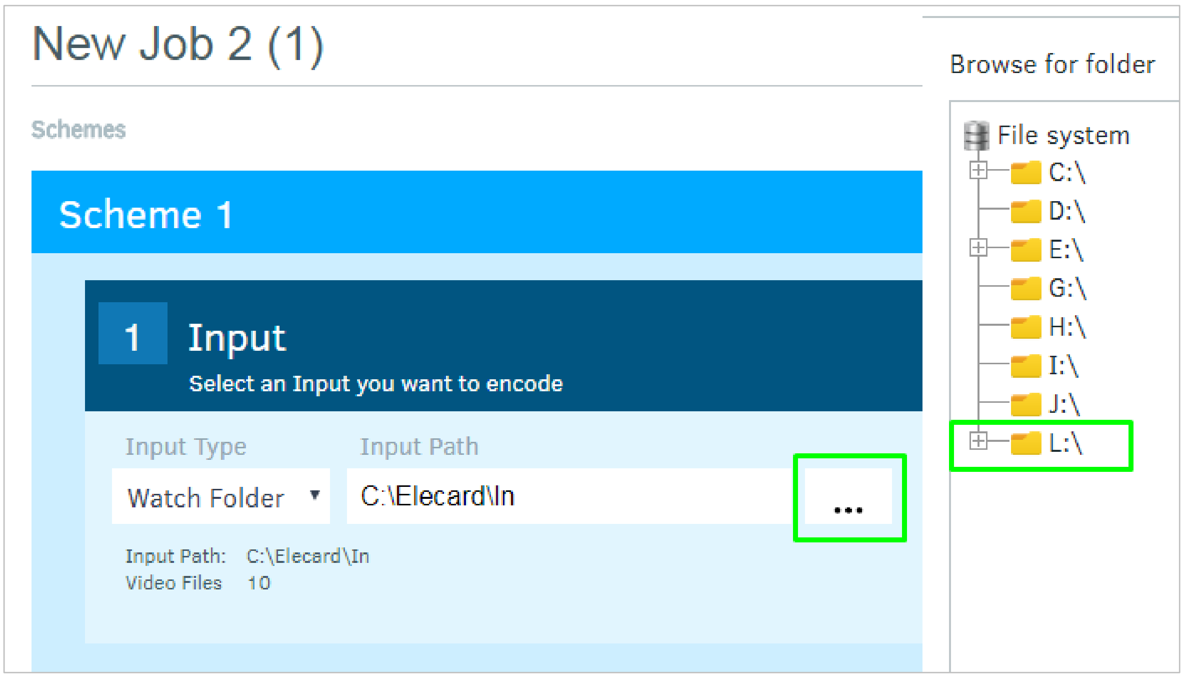


Figure 14. Task Scheduler Library

- To check the drive mounting, go to WEB manager and open the Browse window.



New Job 2 (1)

Schemes

Scheme 1

1 Input

Select an Input you want to encode

Input Type	Input Path
Watch Folder	C:\Elecard\In

Input Path: C:\Elecard\In
Video Files 10

Figure 15. Browse Window

3 Advanced Features

3.1 API

To use API, it is required to go to the address: ip_server_address:9003 over WebSocket by sending a request in xml format.

3.1.1 Presets

There is a list of available presets:

Video

AVC 720 HD (1280x720)
AVC 1080 FHD (1920x1080)
AVC 240 LD (426x240)
AVC 480 SD (854x480)
AVC 360 SD (640x360)
AVC 144 LD (256x144)
AVC 4K UHD (3840x2160)
HEVC 4K UHD (3840x2160)
HEVC 720 HD (1280x720)
HEVC 480 SD (854x480)
HEVC 360 SD (640x360)
HEVC 144 LD (256x144)
HEVC 240 LD (426x240)
HEVC 1080 FHD (1920x1080)
Mpeg2 720 HD (1280x720)
Mpeg2 1080 FHD (1920x1080)
Mpeg2 480 SD (854x480)
Mpeg2 360 SD (640x360)
Mpeg2 144 LD (256x144)
Mpeg2 240 LD (426x240)

Audio:

AAC (1 ch)

Output:

DASH
File Render (mp4 mux)
File Render (ts mux)
HLS

Note: *File Render (mp4 mux) – is MP4 container; File Render (ts mux) – is TS container.*

3.1.2 Request for Creating a Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
    <SetValue Name="WatchFolderConfig">
        <Parameter>
            <p>Start</p>
            <p>Input path</p>
            <p>Output path</p>
            <p></p>
            <p></p>
            <p></p>
            <p>video preset 1</p>
            <p>video preset 2</p>
            <p>video preset n</p>
            ....
            <p>audio preset</p>
            <p>out preset</p>
        </Parameter>
    </SetValue>
</XMLConfig>
```

where:

- Input path – is an input folder,
- Output path – is an output folder,
- <p>video preset n</p> - a list of required presets selected from the list specified in the Section 3.1.1 Presets.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
    <SetValue Name="WatchFolderConfig">
        <RetVal>ID</RetVal>
    </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier for a created task.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Start</p>
                  <p>C:\Elecard\In</p>
                  <p>C:\Elecard\Out</p>
                  <p></p>
                  <p></p>
                  <p></p>
                  <p>AVC 720 HD (1280x720)</p>
                  <p>AAC (1 ch)</p>
                  <p>File Render (ts mux)</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
    <SetValue Name="WatchFolderConfig">
        <RetVal>90eae1d1-45fe-4210-be16-5152829e7be3</RetVal>
    </SetValue>
</XMLConfig>
```

3.1.3 Request for Receiving a List of Tasks

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
    <GetValue Name="WatchFolderConfig">
        <Parameter>List</Parameter>
    </GetValue>
</XMLConfig>
```

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
    <GetValue Name="WatchFolderConfig">
        <RetVal>
            <WatchFolders>
                  <Task>ID 1</Task>
                  <Task>ID 2</Task>
                  <Task>ID n</Task>
            </WatchFolders>
        </RetVal>
    </GetValue>
</XMLConfig>
```

where:

- ID – is a task unique identifier.

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
```

```
<XMLConfig dispatcher="1">
    <GetValue Name="WatchFolderConfig">
        <RetVal>
            <WatchFolders>
                <Task>59a98638-4f3d-4792-aa94-b850497bd2c5</Task>
                <Task>90eae1d1-45fe-4210-be16-5152829e7be3</Task>
            </WatchFolders>
        </RetVal>
    </GetValue>
</XMLConfig>
```

3.1.4 Request for File Processing

Before processing a file, it is required to create a task. In this case it is not required to specify an input folder when creating a task, and the field <p>Input path</p> should be left empty, but not deleted.

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Put</p>
                  <p>ID</p>
                  <p>Input file path</p>
                  <p></p>
                  <p>Time of start in ms</p>
                  <p>Time of stop in ms</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier for the created task,
- Input file path – is a path to a file (or to a network file, e.g.: \\192.168.1.10\elecard\test.mp4).
- Time of start in ms – time of start of a cut out fragment, in milliseconds,
- Time of stop in ms – time of stop of a cut out fragment, in milliseconds.

If cutting out of a file fragment is not required, leave fields of these tags (last three tags starting from the <p></p> tag) empty or delete them completely.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <SetValue Name="WatchFolderConfig">
            <RetVal>
                  <WatchFolder id="ID" />
            </RetVal>
      </SetValue>
</XMLConfig>
```

where:

- ID – is an identifier of the task receiving a request for file processing.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Put</p>
                  <p>90eae1d1-45fe-4210-be16-5152829e7be3</p>
                  <p>C:\Elecard\In\test.mp4</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
    <SetValue Name="WatchFolderConfig">
        <RetVal>
            <WatchFolder id="90eae1d1-45fe-4210-be16-5152829e7be3" />
        </RetVal>
    </SetValue>
</XMLConfig>
```

3.1.5 Request for Sending a File over HTTP or FTP

Before processing a file, it is required to create a task (see the Section 3.1.2 Request for Creating a Task). In this case it is not required to specify an input folder when creating a task, and the field <p>Input path</p> should be left empty, but not deleted. If you are an authorized user, it is required to use the following structure: http://login:password@link_to_file (e.g. http://login:password@elecard.com/test.mp4).

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Download</p>
                  <p>ID</p>
                  <p>link to file</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,
- link to file – is a link to a file.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>ID_download</RetVal>
      </GetValue>
</XMLConfig>
```

where:

- ID_download – is a unique identifier of download.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Download</p>
                  <p>90eae1d1-45fe-4210-be16-5152829e7be3</p>
                  <p>http://elecard.com/test.mp4</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>b86f5fd4-e0d7-4485-94f7-c2363799a0f8</RetVal>
      </GetValue>
</XMLConfig>
```

3.1.6 Request for Information on a File Downloading Process over HTTP or FTP

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Progress</p>
                  <p> ID_download</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID_download – is a unique identifier of download.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
	<GetValue Name="WatchFolderConfig">
		<RetVal>
			<Download>
				<Progress></Progress>
				<Downloaded></Downloaded>
				<Aborted></Aborted>
				<LastError />
				<Done></Done>
			</Download>
		</RetVal>
	</GetValue>
</XMLConfig>
```

where:

- Progress – is a current downloading process of a file,
- Downloaded – the number of downloaded bytes,
- Aborted – aborted download status,
- LastError – an error occurred when downloading a file,
- Done – completed download status.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
	<GetValue Name="WatchFolderConfig">
		<Parameter>
			<p>Progress</p>
			<p>b86f5fd4-e0d7-4485-94f7-c2363799a0f8</p>
		</Parameter>
	</GetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
	<GetValue Name="WatchFolderConfig">
		<RetVal>
			<Download>
				<Progress>100</Progress>
				<Downloaded>5007364</Downloaded>
				<Aborted>0</Aborted>
				<LastError />
				<Done>1</Done>
			</Download>
		</RetVal>
```

```
      </GetValue>
</XMLConfig>
```

3.1.7 Request for a Task Status (Status/Progress/Error)

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Status</p>
                  <p>ID</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>
                  <Task>
                        <Status></Status>
                        <Progress></Progress>
                        <Error></Error>
                  </Task>
            </RetVal>
      </GetValue>
</XMLConfig>
```

where:

- Status – is a task status (waiting, completed, failed),
- Progress – a processing progress of a current file,
- Error – an error text.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Status</p>
                  <p>90eae1d1-45fe-4210-be16-5152829e7be3</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>
                  <Task>
                        <Status>Running</Status>
                        <Progress>14</Progress>
                  </Task>
            </RetVal>
      </GetValue>
</XMLConfig>
```

3.1.8 Request for Receiving ID Download

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Get</p>
                  <p>ID</p>
                  <p>GUIDDownload</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>ID_download</RetVal>
      </GetValue>
</XMLConfig>
```

where:

- ID_download – is a unique identifier of download.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Get</p>
                  <p>90eae1d1-45fe-4210-be16-5152829e7be3</p>
                  <p>GUIDDownload</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal> b86f5fd4-e0d7-4485-94f7-c2363799a0f8</RetVal>
      </GetValue>
</XMLConfig>
```

3.1.9 Request for Receiving a List of Files in a Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher='1' >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Queue</p>
                  <p>ID</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>
                  [Status] fileName
            </RetVal>
      </GetValue>
</XMLConfig>
```

where:

- fileName – is a file name,
- Status – is a file status that can have the following values:
  - Configuring – file preparation for the processing queue,
  - Running – a file is being processed,
  - Success – a file is successfully processed,
  - Waiting – a file is in queue for further processing,
  - Fault – a processing error,
  - Critical Error – a processing fatal error.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
```

```
<XMLConfig dispatcher='1' >
	<GetValue Name="WatchFolderConfig">
		<Parameter>
			<p>Queue</p>
			<p>b7da8473-a19c-4c8d-8f00-7f298d47e9b1</p>
		</Parameter>
	</GetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
	<GetValue Name="WatchFolderConfig">
		<RetVal>
			[Configuring] Tom_and_Jerry_h264_AAC - копия.mp4
			[Waiting] Tom_and_Jerry_h264_AAC.mp4
			[Running] Tom_and_Jerry_h264_AAC - копия - Copy.mp4
			[Success] Tom_and_Jerry_h264_AAC - Copy.mp4
		</RetVal>
	</GetValue>
</XMLConfig>
```

3.1.10 Request for Suspending a Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
	<SetValue Name="WatchFolderConfig">
		<Parameter>
			<p>Suspend</p>
			<p>ID</p>
		</Parameter>
	</SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
	<SetValue Name="WatchFolderConfig">
		<RetVal>ID</RetVal>
	</SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the stopped task.

3.1.11 Request for Resuming a Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Resume</p>
                  <p>ID</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <SetValue Name="WatchFolderConfig">
            <RetVal>ID</RetVal>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the started task.

3.1.12 Request for Deleting a Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Stop</p>
                  <p>ID</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <SetValue Name="WatchFolderConfig">
            <RetVal>ID</RetVal>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the deleted task.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Stop</p>
                  <p>90eae1d1-45fe-4210-be16-5152829e7be3</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <SetValue Name="WatchFolderConfig">
            <RetVal>90eae1d1-45fe-4210-be16-5152829e7be3</RetVal>
      </SetValue>
</XMLConfig>
```

3.1.13 Request for Restarting a Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <SetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Reset</p>
                  <p>ID</p>
                  <p>Input path</p>
                        <p>Output path</p>
                        <p></p>
                        <p></p>
                        <p></p>
                        <p>video preset 1</p>
                        <p>video preset 2</p>
                        <p>video preset n</p>
                        ....
                        <p>audio preset</p>
                        <p>out preset</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,

- Input path – an input folder,
- Output path – an output folder,
- <p>video preset n</p> - a list of required presets selected from the list specified in the Section 3.1.1 Presets.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
	<SetValue Name="WatchFolderConfig">
		<RetVal>ID</RetVal>
	</SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of a task.

An example of a request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
	<SetValue Name="WatchFolderConfig">
		<Parameter>
			<p>Reset</p>
			<p>90eae1d1-45fe-4210-be16-5152829e7be3</p>
			<p>C:\Elecard\In</p>
			<p>C:\Elecard\Out</p>
			<p></p>
			<p></p>
			<p></p>
			<p>AVC 720 HD (1280x720)</p>
			<p>AAC (1 ch)</p>
			<p>File Render (ts mux)</p>
		</Parameter>
	</SetValue>
</XMLConfig>
```

An example of a response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
	<SetValue Name="WatchFolderConfig">
		<RetVal>90eae1d1-45fe-4210-be16-5152829e7be3</RetVal>
	</SetValue>
</XMLConfig>
```

3.1.14 Request for Setting Time for Automatic Deletion of the Completed Task

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
	<SetValue Name="WatchFolderConfig">
```

```
            <Parameter>
                  <p>SetAliveTime</p>
                  <p>ID</p>
                  <p>value</p>
            </Parameter>
      </SetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,
- value – time interval in seconds.

3.1.15 Request for Receiving Information on the Time Remaining before Task Automatic Deletion

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>RestOfTime</p>
                  <p>ID</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>value</RetVal>
      </GetValue>
</XMLConfig>
```

where:

- value – time remaining before task automatic deletion, in seconds.

3.1.16 Request for Storing User Records

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Set</p>
                  <p>ID</p>
                  <p>Key</p>
                  <p>Value</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,
- Key – is a unique key specified by a user,
- Value – a string specified by a user.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>Success</RetVal>
      </GetValue>
</XMLConfig>
```

3.1.17 Request for Obtaining User Records

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Get</p>
                  <p>ID</p>
                  <p>Key</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,
- Key – is a unique key specified by a user.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal> Value</RetVal>
      </GetValue>
</XMLConfig>
```

where:

- Value – a string specified by a user.

3.1.18 Request for Adding an Encoding Node

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Nodes</p>
                  <p>Add</p>
                  <p>Address</p>
                  <p>Port</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- Address – is an IP address of an encoding node that should be added,
- Port – is a port of an encoding node that should be added.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal></RetVal>
      </GetValue>
</XMLConfig>
```

3.1.19 Request for Removing an Encoding Node

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Nodes</p>
                  <p>Remove</p>
                  <p>Address</p>
                  <p>Port</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

where:

- Address – is an IP address of an encoding node that should be removed,
- Port – is a port of an encoding node that should be removed.

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
```

```
      <GetValue Name="WatchFolderConfig">
            <RetVal></RetVal>
      </GetValue>
</XMLConfig>
```

3.1.20 Request for Receiving a List of Encoding Nodes

Request:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1" >
      <GetValue Name="WatchFolderConfig">
            <Parameter>
                  <p>Nodes</p>
                  <p>List</p>
            </Parameter>
      </GetValue>
</XMLConfig>
```

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>
                  <CSNodes>
                  <HOST Address="ADDRESS" Port="PORT" />
                  <HOST Address="ADDRESS" Port="PORT" />
                   ...
                  </CSNodes>
            </RetVal>
      </GetValue>
</XMLConfig>
```

where:

- Address – is an IP address of an encoding node,
- Port – is a port of an encoding node.

3.1.21 Request for Obtaining All Task Statuses

Request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<XMLConfig dispatcher='1' >
      <GetValue Name="WatchFolderConfig">
            <Parameter>StatusAll</Parameter>
      </GetValue>
</XMLConfig>
```

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
```

```
      <GetValue Name="WatchFolderConfig">
            <RetVal>
                  <WatchFolderConfig>
                        <WatchFolder id="ID">
                              <Task Status="STATUS" />
                        </WatchFolder>
                  </WatchFolderConfig>
            </RetVal>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,
- Status – is a task status (waiting, completed, failed).

3.1.22 Request for Obtaining All Task Parameters

Request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<XMLConfig dispatcher='1' >
      <GetValue Name="WatchFolderConfig">
            <Parameter>ConfigAll</Parameter>
      </GetValue>
</XMLConfig>
```

Response:

```
<?xml version="1.0" encoding="windows-1251" ?>
<XMLConfig dispatcher="1">
      <GetValue Name="WatchFolderConfig">
            <RetVal>
                  <WatchFolderConfig>
                        <WatchFolder id="ID" input="INPUT" output="OUTPUT">
                              <Profile>PROFILE</Profile>
                              ...
                              <Consoles>
                                    <Host Name="NAME" Consoles="CONSOLES" />
                                    ...
                              </Consoles>
                              <Property key="KEY" value="VALUE" />
                              ...
                        </WatchFolder>
                        ...
                  </WatchFolderConfig>
            </RetVal>
      </GetValue>
</XMLConfig>
```

where:

- ID – is a unique identifier of the created task,

- input – is an input folder,
- output – is an output folder,
- Profile – is an encoding profile,
- Name – is a server name,
- Consoles – is a list of the server consoles,
- key – is a unique key specified by a user,
- value – is a string specified by a user.