REST documentation for MagicSoft Recorder

Description

By using the MagicSoft Recorder Web server you can remotely get the status of the recorder channels, control them and get a visual and audio stream of what the channel streams. This functionality is actually split into two logical servers, both existing inside the same webserver executable: a http server and a stream server.

HTTP Server

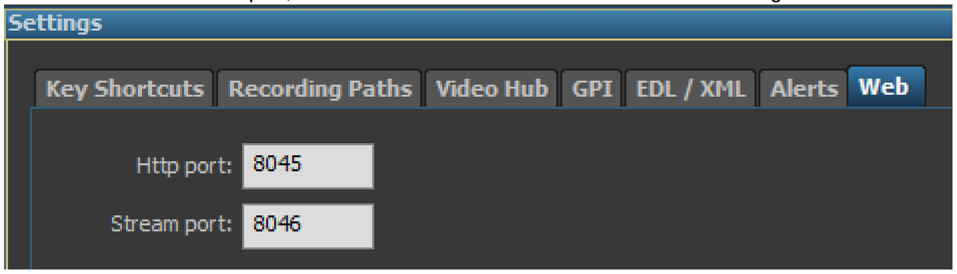
- Is the part that contains most of the functionality of the web server
- Uses REST messages to query and control the status of the recorder channels
- Responds to usual file GET messages in order to return the needed html / css / js files to show the web interface

Stream Server

- Uses web sockets to stream video and audio data

- Uses a different port than the HTTP Server
- Clients can register and unregister to a channel's stream through REST messages sent to the HTTP Server

Each server uses it's own port, which can be modified in the Recorder Web Settings:



Http port: 8045

Stream port: 8046

REST Messages

This is a detailed list of the REST messages that the Recorder Webserver contains. Remember, they can only be sent to the Http Server.

Note: [boolean] has a value of either 0 (for false) or 1 (for true)

program/status

- Description: Used to get information about the MagicSoft Recorder instance that the server is attached to.
- Needs Web License: No
- Verb: GET
- Parameters: none
- Response:

```
{
    http_port: [number],     // port of the Http Server as saved in the Recorder Settings
    stream_port: [number],   // port of the Stream Server as saved in the Recorder Settings
    license_web: [boolean],  // sends if the Recorder has Web License
    license_edl: [boolean],  // sends if the Recorder has Edl License
    language:
    {
        name: [string],      // the name of the Recorder's current language (e.g. French)
        url: [string]        // the path to the current language file (e.g.
language/French.loc)
    },
    success: [boolean],       // if the command was a success
    //if not successful:
    error:
    {
        code: [number],       // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: program/status // for convenience
}
```

recording/status

- Description:Get information about the status of one or all channels.
- Needs Web License: No
- Verb: GET
- Parameters:
    - channel=[number] *//The channel for which you want to receive information.*
        - If between 0 and 3, it will send information about one channel
        - Otherwise, it will send information about all channels
- Response:

```
{
    status: // if multiple channels are requested, this is actually an array of structures
    {
        enabled: [boolean],   // if the channel is enabled or not in the Recorder
        remote: [boolean],    // if the channel is currently being controlled through web
        recording: [boolean], // if the channel is currently recording
        forbidden: [boolean], // if the channel is currently forbidden to be controlled
through web (can be changed in the Recorder Web Settings)
        controller: [string], // the name of the current controller (ip or hostname)
        video_mode: [number], // the current video mode of the channel [-1 = Unkown, 0 =
PAL, 1 = NTSC, 2 = HD_720p_50, 3 = HD_720p_59_94, 4 = HD_1080i_50, 5 = HD_1080i_59_94]
        name: [string],       // the channel's name
        status_text: [string], // string representing the current status of the channel
        presets: [array of strings] // array of presets the channel currently has
        selected_preset: [array of numbers] //array of indices in the presets array
representing the selected preset for each encoder
        time_elapsed: [number], // number of seconds elapsed since the recording started
        time_remaining: [number], // number of seconds remaining until the recording ends
        video_hub_channels: [array of strings], //array of channels present in the Video Hub
        video_hub_selected: [number], // index in the video_hub_channels array representing
the selected Video Hub channel
        marks_count: [number], // the number of marks
    },
    channel: [number],        // the channel that the info was requested for
    success: [boolean],       // if the command was a success
    //if not successful:
    error:
    {
        code: [number],       // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/status // for convenience
}
```

recording/rec

- Description: Start the recording on the requested channel
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - channel=[number] *//The channel for which you want to start the recording [0..3]*
    - recname=[string] *//The name of the recording*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/rec // for convenience
}
```

recording/stop

- Description: Stop the recording on the requested channel
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - channel=[number] *//The channel for which you want to stop the recording [0..3]*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/stop // for convenience
}
```

recording/split

- Description: Split the recording on the requested channel
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - channel=[number] *//The channel for which you want to split the recording [0..3]*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/split // for convenience
}
```

recording/mark

- Description: Mark the position on the recording for the requested channel
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - channel=[number] *//The channel for which you want to mark the recording [0..3]*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/mark // for convenience
}
```

recording/preset

- Description: Changes the preset on the requested channel
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - channel=[number] *//The channel for which you want to change the preset [0..3]*
    - encoder=[number] *//The encoder for which to change the preset (0 or 1)*
    - videomode=[number] *//The video mode for which to change the preset (-1 = Unkown, 0 = PAL, 1 = NTSC, 2 = HD_720p_50, 3 = HD_720p_59_94, 4 = HD_1080i_50, 5 = HD_1080i_59_94)*
    - preset=[number] *//The index from the presets array to use (see the recording/status message)*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/preset // for convenience
}
```

recording/time/add

- Description: Adds time to the current recording
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - channel=[number] *//The channel for which you want to add time [0..3]*
    - time=[number] *//The number of seconds to add*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: recording/time/add // for convenience
}
```

stream/subscribe

- Description: Subscribes the client to the stream of video and audio packages for the requested channel. The stream is available through the web socket opened on the Stream Server.
- Needs Web License: No
- Verb: POST
- Parameters:
  - channel=[number] *//The channel for which you want to subscribe to stream [0..3]*
  - sample_rate=[number] *//The sample rate requested for audio packages*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: stream/subscribe // for convenience
}
```

stream/unsubscribe

- Description: Unsubscribes the client from the stream of video and audio packages for the requested channel.
- Needs Web License: No
- Verb: POST
- Parameters:
  - channel=[number] *//The channel for which you want to unsubscribe from stream [0..3]*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: stream/unsubscribe // for convenience
}
```

stream/package

- Description:This is a REST message that is sent via a web socket by the Stream server to the client containing video and audio packages. This message should not be sent directly by the client to the server! Instead, use the **stream/subscribe** message.
- Needs Web License: No
- Verb: NONE
- Parameters: none
- Response:

```
{
    video: [encoded data],     // base64-encoded png image representing the current video
frame
    samples_count:[number],    // the number of audio samples in the package
    audio_ch1: [encoded data], // base64-encoded audio data for channel 1
    audio_ch2: [encoded data], // base64-encoded audio data for channel 2
    audio_ch1_level: [number], // audio level for channel 1 [-36..0]
    audio_ch2_level: [number], // audio level for channel 2 [-36..0]
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: stream/package // for convenience
}
```

videohub/change

- Description: Used to change the current channel in Video Hub
- Needs Web License: Yes
- Verb: POST
- Parameters:
    - index=[number] *//The index from the Video Hub channels array to change the selection to (see list in recording/status message)*
- Response:

```
{
    channel: [number],       // the requested channel
    success: [boolean],      // if the command was a success
    //if not successful:
    error:
    {
        code: [number],      // see Errors chapter
        message: [string]    // see Errors chapter
    },
    request_url: videohub/change // for convenience
}
```

Errors

If a command isn't successful, you will get an error structure inside the response object, containing a code and a message. The list of error messages is as follows:

- "**Invalid arguments**"
  - Code: 1
  - Description: The rest message was sent with an invalid argument. Check if the parameters are within expected boundaries
- "**Channel is unavailable**"
  - Code: 2
  - Description: The current channel is either forbidden for remote control, or is currently being controlled by someone else
- "**No license detected**"
  - Code:3
  - Description: The MagicSoft Recorder instance to which the server is connected, doesn't have a Web License and has limited functionality.
- "**Could not process request**"
  - Code:4
  - Description: Something happened that didn't allow the message to be processed.