SOUND4 Stream Library [1.1.0]

Generated by Doxygen 1.9.1

1 SOUND4 Stream Library 1

1.1 General usage 1

1.2 Installation 1

1.3 Integrators 1

1.4 Setup Parameters 1

1.5 Metadata 2

1.6 Metadata XML format 2

2 File Index 3

2.1 File List 3

3 File Documentation 5

3.1 documentation.h File Reference 5

3.2 documentation.h 5

Index 7

Chapter 1

SOUND4 Stream Library

1.1 General usage

This library can be used with SOUND4 .CL HD and WEB processing families. This extension supports Live processing only.

1.2 Installation

You just have to put this library in the same directory of the process, and it will detect and use it, if possible.

You can check in the About window, it should appear with its version.

You will need a `Stream.CL` license activated by SOUND4.

You can put a folder named "SOUND4 Stream.CL Metaparsers" in the same directory, which contains some `Lua` files with predefined metadata parser scripts.

1.3 Integrators

When integrating in your software, like radio automation software, you can automatically set the metadata from it, using its SetMetadata function.

You should set the `Metadata_no_ui` before startup to avoid that the user can manage it on his side (Setup Parameters).

1.4 Setup Parameters

Those parameters can be send to the process at startup by using its `SetParameter` function before creating the instance. They will configure what will be available to the user in the interface :

- `Metadata_no_ui` : If set, disable the user access to metadata (0/1)
- `Metadata_provided_path` : A directory were there are software provided metadata scripts (.lua). If not set, will search in directory "SOUND4 Stream.CL Metaparsers" located where the library is.
- `Metadata_no_auto_path` : If set, disable the search of scripts in "SOUND4 Stream.CL Metaparsers" (0/1)
- `Metadata_server_maxclient` : Set the maximum count of client to metadata TCP server (default: 1)

1.5 Metadata

Metadata can be used to set the stream artist/title/song/... information that will be passed to the streaming servers.

The following metadata can be passed :

- `title` : The title of the current song
- `artist` : The name of the artist of the current song
- `compact_song` : The compact_song sent to stream servers using single metadata info. Automatically built from artist and title if not set
- `xml_metadata` : The metadata in XML form. See Metadata XML format

1.6 Metadata XML format

The metadata XML should have the following format (each is optional):

```
<?xml version="1.0" encoding="UTF-8" ?>
<metadata>
     <title>Strange Fruit</title>
     <artist>Billie Holiday</artist>
     <!-- compact_song will be made automatically with artist - title if not specified -->
     <compact_song>Billie Holiday - Strange Fruit</compact_song>
     <_ice>
         <!-- optional Icecast specific -->
         <title>Strange Fruit</title>
         <artist>Billie Holiday</artist>
     </_ice>
     <_shout>
         <!-- optional Shoutcast specific -->
         <TPE1>Billie Holiday</TPE1>
         <TIT2>Strange Fruit</TIT2>
         <TCON v1="24">Subgenre</TCON>
         <extension>
             <soon>The Next Artist - The Next Title</soon>
         </extension>
     </_shout>
     <_id3>
         <!-- optional ID3 format (used in MPEG-TS) -->
         <TPE1>Billie Holiday</TPE1>
         <TIT2>Strange Fruit</TIT2>
         <TCON>Rock</TCON>
     </_id3>
     <_rtmp>
         <!-- optional RTMP specific -->
         <!-- this will add a field with key "mycommand" in the onMetadata notify AMF block -->
         <mycommand>Send command</mycommand>
     </_rtmp>
</metadata>
```

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

documentation.h 5

Chapter 3

File Documentation

3.1 documentation.h File Reference

3.2 documentation.h

```
00001
```

Index

documentation.h, 5