XEngine\_StreamMedia Service Docment

目录

[XEngine\_StreamMedia Service Docment 1](#_Toc10152)

[Preface 4](#_Toc24747)

[Reader 4](#_Toc28890)

[Overview 4](#_Toc3547)

[Related modules 4](#_Toc4254)

[一 Technical structure 4](#_Toc29617)

[1.1 Instructions 4](#_Toc24147)

[1.1 Stream Protocol 4](#_Toc8464)

[1.2.1 Publish Straem 4](#_Toc22112)

[1.2.2 Play Protocol 5](#_Toc28041)

[二 Configure Environment 5](#_Toc13239)

[2.1 Windows 5](#_Toc27486)

[2.2 LINUX 5](#_Toc24128)

[2.3 MacOS 5](#_Toc20169)

[三 Interface Protocol 6](#_Toc7570)

[3.1 X Stream Protocol 6](#_Toc7223)

[3.3.1 Craete Stream 6](#_Toc15033)

[3.3.2 Destory Stream 7](#_Toc15764)

[3.3.3 Push Stream 8](#_Toc20687)

[3.3.4 Pull Stream 8](#_Toc24297)

[3.3.5 Notify Protocol 9](#_Toc19376)

[3.2 JT1078 Stream Protocol 10](#_Toc13514)

[3.3 GB28181 Stream Protocol 10](#_Toc19741)

[3.4 Convert Push Stream Protocol 10](#_Toc9074)

[3.4.1 Play 10](#_Toc13306)

[3.4.2 Stop 11](#_Toc13180)

[3.4.3 Enum 11](#_Toc7831)

[3.5 HTTP Management API 12](#_Toc27029)

[3.5.1 Get Publish List 12](#_Toc18463)

[3.5.2 Get Pull List 13](#_Toc163)

[四 Configure Description 13](#_Toc25251)

[4.1 Service Configure 13](#_Toc16941)

[4.1.1 basic configure 13](#_Toc23133)

[4.1.2 Max Configure 14](#_Toc22289)

[4.1.3 Time Configure 14](#_Toc5146)

[4.1.4 Pull Configure 14](#_Toc31426)

[4.1.5 Log Configure 14](#_Toc2032)

[五 Advanced instructions 15](#_Toc3495)

[5.1 GOP Cache 15](#_Toc25258)

[5.2 First Play 15](#_Toc10074)

[5.3 Stream Convert 15](#_Toc27537)

[5.4 M3U8 15](#_Toc4041)

[appendix 15](#_Toc6521)

[Appendix update log 15](#_Toc32401)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine\_StreamMedia Service Docment | |
| Be A Version： | V2.6.0.1001 | |
| Released： | 2024-02-03 | |
| Writer： qyt | | |

# Preface

## Reader

Development ,tester,qa

## **Overview**

This document contains related technical descriptions and interface definitions

## Related modules

This service uses XEngine as the development package. To use this service code, the XEngine development environment must be configured and installed.

# 一 Technical structure

The Service implemented through c/c++,Protocol use to tcp.

## Instructions

You can use OBS or FFMPEG to push the stream, currently only support H264 video, audio AAC, H265 video will be supported later

Srt publish stream cloud be publish h264 or h265.

## Stream Protocol

### 1.2.1 Publish Straem

RTMP Publish:rtmp://127.0.0.1/live/qyt

SRT Publish:srt://127.0.0.1:5603?streamid=#!::h=live/qyt,m=publish

XStream Publish:need to private protocol

JT1078 Publish:Just accept the 1078 data stream, do not process the command, and set it to our server through the command

### 1.2.2 Play Protocol

FLV OVER HTTP:http://127.0.0.1:5600/api?stream=play&sms=live/qyt&type=flv

TS OVER HTTP:http://127.0.0.1:5600/api?stream=play&sms=live/qyt&type=ts

RTSP Play:rtsp://127.0.0.1:5600/api?stream=play&sms=live/qyt&type=rtsp

RTMP Play:rtmp://127.0.0.1/live/qyt

SRT Play:srt://127.0.0.1:5603?streamid=#!::r=live/qyt,m=request

XStream Over HTTP:http://127.0.0.1:5601?api?stream=play&sms=live/qyt&type=xstream

# 二 Configure Environment

### 2.1 Windows

When you complete with configuration.you can come in code path.open XEngine.sln by vs

If environment not have error.complie is succesed

And you need copy file under XEngine\_Release to your complied dir.next step copy file under XEngine depend module to your complied dir.

Note: You can run the program directly, the system will prompt you what you need, you can directly enter the XEngine directory to search,you can also use vscopy-\*.bat to copy dependent module.

## 2.2 LINUX

If you use linux.you must running on ubuntu22.04 or RockyLinux9.x(Compatible with Centos-like systems)...

## 2.3 MacOS

MacOS support 13.x system,just x64 bit.not support m1

# 三 Interface Protocol

For specific protocols, please refer to the header file:XEngine\_UserProtocol.h

## 3.1 X Stream Protocol

### 3.3.1 Craete Stream

The TOKEN field for creating a stream protocol is optional. If filled in, it indicates the unique ID of the stream, so that users can clearly know their push stream ID. For example, if the stream ID is 123, then your push address is: http://192.168.1.10 /live/123

#### 3.3.1.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQCREATE

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

push stream addresss by tszSMSAddr

#### 3.3.1.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REPCREATE

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

//The reply content is the request content

### 3.3.2 Destory Stream

#### 3.3.2.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQDESTROY

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

XENGINE\_PROTOCOLSTREAM

#### 3.3.2.2 Reply

Reply only when reply is 1

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REPDESTROY

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

//The reply content is the request content

### 3.3.3 Push Stream

#### 3.3.3.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQPUSH

unPacketSize = sizeof(XENGINE\_PROTOCOLDATA) + AVDATA

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:AV Code Datas

#### 3.3.3.2 Reply

none

### 3.3.4 Pull Stream

#### 3.3.4.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = Stream Token

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQPULL

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

XENGINE\_PROTOCOLSTREAM

#### 3.3.4.2 Reply

The wReserver field is 0 means success, and other values is failure. The failed values are as follows:

1:not found stream id

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = Stream Token

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REPXPULL

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

//The reply content is the request content

### 3.3.5 Notify Protocol

The notification protocol is used to handle the push-pull state, inform the other party of the current status of receiving and processing data, and let the sender or receiver adjust the sending frequency!

The pull-stream protocol is the same as the push-stream protocol, but the CODE value is different, which will not be demonstrated in the following.

Also note: the push stream protocol is not used for pull stream, similarly, the pull stream notification protocol cannot be used for push stream

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = Stream Token

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_STATPUSH

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM) + sizeof(XENGINE\_SMSPROTOCOL)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:sizeof(XENGINE\_PROTOCOLSTREAM) + sizeof(XENGINE\_SMSPROTOCOL)

## 3.2 JT1078 Stream Protocol

Refer to the JT1078-2014 (2016) document customized for the platform of the Ministry of Communications

## 3.3 GB28181 Stream Protocol

Refer GB/T28181-2016 Docment

## 3.4 Convert Push Stream Protocol

Token is control by yourself or system create,when you stop this stream need this token.otherwise this stream is always exist.

The Protocol Applies to XEngine\_ForwardApp

### 3.4.1 Play

#### 3.4.1.1 Request

Interface:HTTP

Method:POST

Address:http://127.0.0.1:5602/api?function=play&token=100010001

Parament:function:Operator Method

Token:Stream Number

Payload:RTSP or other protocol play address

{

"tszAVUrl":"rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny\_115k.mp4"

}

#### 3.4.1.2 Reply

{

"code":0,

"msg":"success",

"token":"100010001"

}

### 3.4.2 Stop

#### 3.4.2.1 Request

Interface:HTTP

Method:POST

Address:http://127.0.0.1:5602/api?function=stop&token=10001001

Parament:function:Operator Method

Token:Stream Number

#### 3.4.2.2 Reply

{

"code":0,

"msg":"success"

}

### 3.4.3 Enum

Can be enum all play list

#### 3.4.3.1 Request

Interface:HTTP

Method:GET

Address:http://127.0.0.1:5602/api?function=list

Parament:function:Operator Method

#### 3.4.3.2 Reply

{

"Array":[

{

"tszToken":"100010001",

"tszAVUrl":"rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny\_115k.mp4"

}

],

"Count":1,

"code":0,

"msg":"success"

}

## 3.5 HTTP Management API

### 3.5.1 Get Publish List

#### 3.5.1.1 Request

Interface:HTTP

Method:GET

Address:http://127.0.0.1:5600/api?function=publish&method=get

#### 3.5.1.2 Reply

{

"Array":[

{

"enStreamType":3,

"nClientCount":0,

"st\_AudioInfo":{

"bEnable":true,

"enAVCodec":10,

"nBitRate":160,

"nChannel":2,

"nSampleFmt":16,

"nSampleRate":48000

},

"st\_VideoInfo":{

"bEnable":true,

"enAVCodec":7,

"nBitRate":5950,

"nFrameRate":30,

"nHeight":1080,

"nWidth":1920

},

"tszSMSAddr":"live/qyt"

}

],

"code":0,

"msg":"success"

}

### 3.5.2 Get Pull List

#### 3.5.2.1 Request

Interface:HTTP

Method:GET

Address:http://127.0.0.1:5600/api?function=pull&method=get

#### 3.5.2.2 Reply

{

"Array":[

{

"enStreamType":17,

"tszPushAddr":"127.0.0.1:50795",

"tszSMSAddr":"live/qyt"

}

],

"code":0,

"msg":"success"

}

# 四 Configure Description

## 4.1 Service Configure

Basic Configure File:XEngine\_Config.json

### 4.1.1 basic configure

* tszSMSUrl:Push Address
* tszIPAddr: location ip address
* bDeamon: 1 deamon process run 0 is terminal run
* nRTMPPort:RTMP protocol port
* nHttpPort:HTTP Port
* nXStreamPort:tcp port
* nJT1078Port:JT 1078 port
* nSrtPort:srt Port

### 4.1.2 Max Configure

XMax Configure

* nMaxClient Allow Max Client Count
* nMaxQueue Allow Max Queue
* nIOThread:network io process threads number
* nHttpThread:http process threads number
* nXStreamThread:XStream process threads number
* nRTMPThread:RTMP Process Threads number
* nJT1078Thread:1078 Process Threads Number

### 4.1.3 Time Configure

XTime Configure

* nTimeCheck:check time
* nHttpTimeOut:same nTCPTimeOut
* nXStreamTimeOut:XStream Timeout
* nRTMPTimeout:RTMP Client Timeout
* nJT1078Timeout:1078 Client Timeout

### 4.1.4 Pull Configure

XPull Configure

* XStream:Whether Enable to Xstream Supported
* RTMP:Whether Enable to RTMP Supported
* FLV:Whether Enable to flv Supported
* RTSP:Whether Enable to rtsp Supported
* RTC:Whether Enable to rtc Supported,并且设置绑定的端口
* SRT:Whether Enable to srt Supported
* TS:Whether Enable to ts over http Supported
* HLS:Whether Enable to m3u8 pull Supported,and configure to clear and time and save path

### 4.1.5 Log Configure

XLog Configure

* MaxSize:Log file size
* MaxCount:Log File Number
* LogLeave:Allow save level

# 五 Advanced instructions

## 5.1 GOP Cache

The GOP cache is not currently supported, so there will be a black screen problem when the client streams and plays. When an I frame is encountered, it can be played normally.

In other words, there is currently no delay.

GOP buffering will be implemented in future

## 5.2 First Play

First play does not suppoted.we will be implemented in future

## 5.3 Stream Convert

As one of the important functions of XStream, stream conversion is of great significance. It can convert push streaming protocols such as SRT, XSTREAM, and RTMP into other pull streaming protocols. For example, they can all be converted into Wait, it is convenient for users to deal with the problem of push and pull streaming protocol restrictions

## 5.4 M3U8

Currently, the streaming media server does not support direct M3U8 pull, only support conversion. You can use third-party tools such as NGINX to pull streams.

The duration set to 15 seconds is not necessarily 15 seconds, the key is the position of the key frame. Generally, it may be extended by a few seconds

# appendix

## Appendix update log