XEngine Proxy Service Docment

目录

[XEngine Proxy Service Docment 1](#_Toc4332)

[Preface 3](#_Toc25037)

[Reader 3](#_Toc10226)

[Overview 3](#_Toc25944)

[Associate Module 3](#_Toc3862)

[一 Technical structure 3](#_Toc30653)

[1.1 Directory Structure 3](#_Toc10514)

[1.2 Source Directory 4](#_Toc15463)

[二 Configure Env 4](#_Toc4309)

[2.1 WINDOWS 4](#_Toc6901)

[2.2 LINUX 4](#_Toc3659)

[2.3 Version Requirements 5](#_Toc18268)

[2.3.1 System Version 5](#_Toc19040)

[2.3.2 Software Version 5](#_Toc16480)

[三 Configure Description 5](#_Toc1068)

[3.1 Basic Configure 5](#_Toc7828)

[3.2 Max Configure 5](#_Toc22632)

[3.3 Time Configure 6](#_Toc8310)

[3.4 Log Configure 6](#_Toc30106)

[四 Interface Protocol 6](#_Toc1135)

[4.1 Tunnel 6](#_Toc10911)

[4.2 Socks 6](#_Toc1972)

[4.3 Forwad 6](#_Toc27254)

[4.3.1 Login 6](#_Toc22881)

[4.3.2 List 7](#_Toc30587)

[4.3.3 Bind 8](#_Toc2889)

[4.4.4 Forwardding 9](#_Toc14714)

[appendix 9](#_Toc5702)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Proxy Service Docment | |
| Be A Version： | V1.4.0.1001 | |
| Released： | 2022-06-17 | |
| Writer： qyt | | |

# Preface

## Reader

Developer ,tester

## **Overview**

This document contains related technical descriptions and interface definitions

## Associate Module

The service used XEngine as Network Toolki.if you want to use code,you have to installed XEngine

Download Address:https://gitee.com/xyry/libxengine

<https://github.com/libxengine/xengine>

Please refer to the readme configuration environment

# 一 Technical structure

Used to tcp protocol though basic protoco

Should be bind two port when Start service

## Directory Structure

* XEngine\_APPClient:example code path
* XEngine\_Docment:docment path
* XEngine\_Release:complie release path
* XEngine\_Source:code path

## Source Directory

* XEngine\_ModuleAuthorize:User authorize module
* XEngine\_ModuleConfigure:configure file module
* XEngine\_ServiceApp:service program
* XEngine\_ThirdPart:3rd-part library

# 二 Configure Env

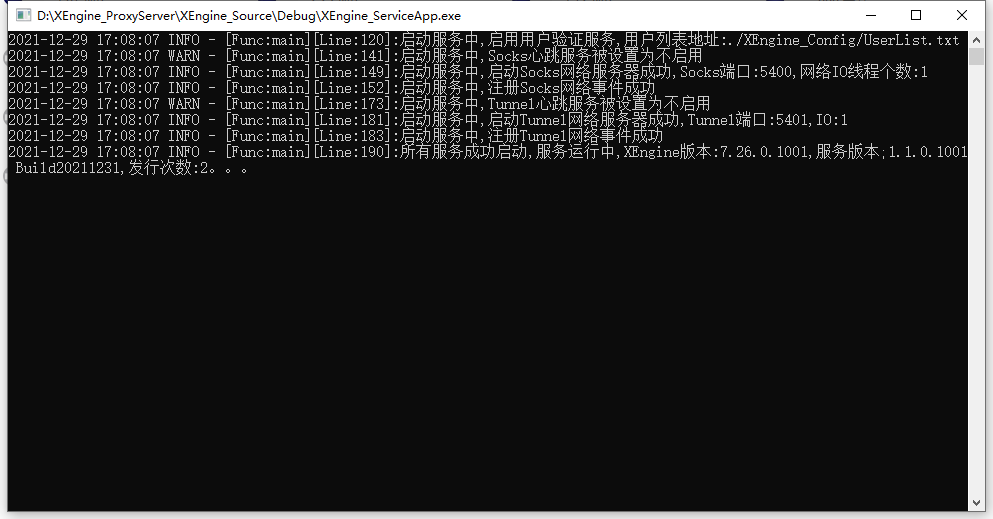
## 2.1 WINDOWS

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

If environment not have error.complie is succesed.contrain 2 module and 1 exe program

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

If not have error,you can see follow the infomation



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-\* scripts to copy dependent modules.

## 2.2 LINUX

Configure complete.you can complie it.open terminal in you source dir and execute command.

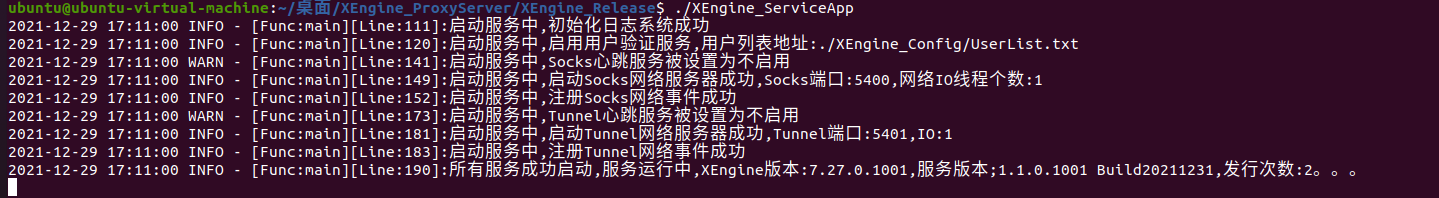
complie:make

install:make FLAGS=InstallAll

clean:make FLAGS=CleanAll

If there is no error.you can see complied XEngine\_ServiceApp file in XEngine\_Release

You can running at terminal.if there is no error,you can infomation:



## 2.3 Version Requirements

### 2.3.1 System Version

Minimum version requirements:

WINDOWS: win7 sp1

Ubuntu:20.04

Centos:8.x

### 2.3.2 Software Version

Minimum version requirements:

XEngine:V7.27

# 三 Configure Description

Configure File:XEngine\_Config.json

## 3.1 Basic Configure

* tszIPAddr:Service IP Address
* bDeamon:Is Deamon running
* nSocksPort:SOCKS Proxy Port
* nTunnelPort:HTTP Proxy Port

## 3.2 Max Configure

XMax Configure

* MaxClient Allow Max Client Count
* MaxQueue Allow Max Queue
* IOThread:network io process threads number

## 3.3 Time Configure

XTime Configure

* nTimeCheck:check time
* nSocksTimeOut:Socks Heartbeat timeout
* nTunnelTimeOut:Tunnel Heartbeat timeout

## 3.4 Log Configure

Configure Information:XLog

* nMaxSize:max log file size
* nMaxCount:file back number
* nLogLeave:log level
* tszLogFile:log save address

# 四 Interface Protocol

## 4.1 Tunnel

TUNNEL standard protocol based on HTTP, please refer to related RFC documents

## 4.2 Socks

Standard protocol based on SOCKSV5, please refer to related RFC documents

## 4.3 Forwad

The forward protocol is a private custom protocol, through which users and lists that can be forwarded can be obtained, and data can be forwarded

### 4.3.1 Login

Can't get list information without logging in

#### 4.3.1.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_AUTH*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LOGREQ

unPacketSize = 0

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 4.3.1.2 Reply

Protocol Header:wReserve 0 is sucesss

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_AUTH*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LOGREP

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 4.3.2 List

#### 4.3.2.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_USER\_FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LISTREQ

unPacketSize = 0

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 4.3.2.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_USER\_FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LISTREP

unPacketSize = JSONSIZE

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

{  
    **"Count"**:**2**,  
    **"Array"**:[  
        **"192.168.1.5:1000"**,  
        **"192.168.1.6:1001"**  
    ]  
}

### 4.3.3 Bind

After binding a forwarding client to request, the other end will also receive the same request. The request only Protocol Header.

#### 4.3.3.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_USER\_FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_BINDREQ

unPacketSize = JSONSIZE

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

{  
    **"tszDstAddr"**:**"192.168.1.6:1000"**  
}

#### 4.3.3.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_*FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_BINDREP

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 4.4.4 Forwardding

If the binding is successful, future data will be forwarded to the corresponding client

# appendix