XEngine Proxy Service Docment

目录

[XEngine Proxy Service Docment 1](#_Toc16320)

[Preface 3](#_Toc24449)

[Reader 3](#_Toc18562)

[Overview 3](#_Toc3554)

[Associate Module 3](#_Toc28938)

[一 Technical structure 3](#_Toc20585)

[1.1 Directory Structure 3](#_Toc28604)

[1.2 Source Directory 3](#_Toc4514)

[1.3 Detailed Protocol 4](#_Toc17345)

[二 Configure Env 4](#_Toc24477)

[2.1 WINDOWS 4](#_Toc18514)

[2.2 LINUX 4](#_Toc8329)

[2.3 MacOS 4](#_Toc19277)

[三 Configure Description 4](#_Toc12665)

[3.1 Basic Configure 4](#_Toc2431)

[3.2 Max Configure 5](#_Toc26619)

[3.3 Time Configure 5](#_Toc18002)

[3.4 Log Configure 5](#_Toc30110)

[3.5 Report Configure 5](#_Toc8629)

[四 Interface Protocol 5](#_Toc25774)

[4.1 Tunnel 5](#_Toc9849)

[4.2 Socks 6](#_Toc24480)

[4.3 Forwad 6](#_Toc7299)

[4.3.1 Login 6](#_Toc16644)

[4.3.2 List 7](#_Toc17780)

[4.3.3 Bind 8](#_Toc6143)

[4.3.4 Forwardding 9](#_Toc8124)

[Appendix 9](#_Toc28056)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Proxy Service Docment | |
| Be A Version： | V1.10.0.1001 | |
| Released： | 2024-11-15 | |
| 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

Please refer to the readme configuration environment

# 一 Technical structure

Used to tcp protocol though basic protoco

Should be bind 3 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\_ModuleConfigure:configure file module
* XEngine\_ModuleProtocol:protocol parse and packet module
* XEngine\_ModuleSession:session handle module
* XEngine\_ServiceApp:service program
* XEngine\_ThirdPart:3rd-part library

## Detailed Protocol

The TCP private protocol adopts the protocol defined by XEngine. You can refer to the protocol document of XEngine to learn more about the definition and interpretation of the protocol

# 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 4 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.

Note: Xengine environment can be copied to your compilation directory through vscopy script, provided that you configure your xengine environment

## 2.2 LINUX

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

## 2.3 MacOS

MacOS requires 13 or above versions, and the compilation and operation mode can refer to linux

# 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
* nForwardPort:forward proxy port

## 3.2 Max Configure

XMax Configure

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

## 3.3 Time Configure

XTime Configure

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

## 3.5 Report Configure

XReport Configure

* bEnable:whether to enable
* tszServiceName:serivce name
* tszAPIUrl:API 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

The forwarding protocol supports both anonymous and named forwarding. The differences between them are as follows:

* Named forwarding requires both parties to undergo login verification before forwarding can be carried out.
* Anonymous forwarding is a connection initiated by the forwarding party. It requires the other party to provide a service. Then, after the requesting party successfully connects, it will perform data forwarding.

### 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 = sizeof(*XENGINE\_PROTOCOL\_USERAUTH*)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

User Info

#### 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:

{

"Array":[

{

"bForward":false,

"st\_UserAuth":{

"enClientType":0,

"enDeviceType":0,

"tszDCode":"",

"tszUserName":"test",

"tszUserPass":""

},

"tszDstAddr":"",

"tszSrcAddr":"127.0.0.1:64900"

}

],

"Count":1

}

### 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.

Anonymous forward does not send request to server

#### 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\_NAMEDREQ

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\_NAMEDREP

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 4.3.4 Forwardding

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

# Appendix