

## 《DDS Windows库编译步骤》

- 1 编译环境
- 2 生成库
  - 2.1 Asio-1.18.2
  - 2.2 OpenSSL
  - 2.3 Memory-0.7-1
  - 2.4 tinyclang
  - 2.5 Fast-CDR
  - 2.6 Fast-DDS
- 3 测试DDS

# 1 编译环境

---

- Visual studio 2017 采用x64 debug进行编译静态库
- cmake

## 2 生成库

---

### 2.1 Asio-1.18.2

---

无需编译，在编译DDS项目时指明头文件路径即可

Asio 是一个跨平台的C开发包用来处理网络和低级I/O编程，通过先进的C方法为开发人员提供连续异步模型。

### 2.2 OpenSSL

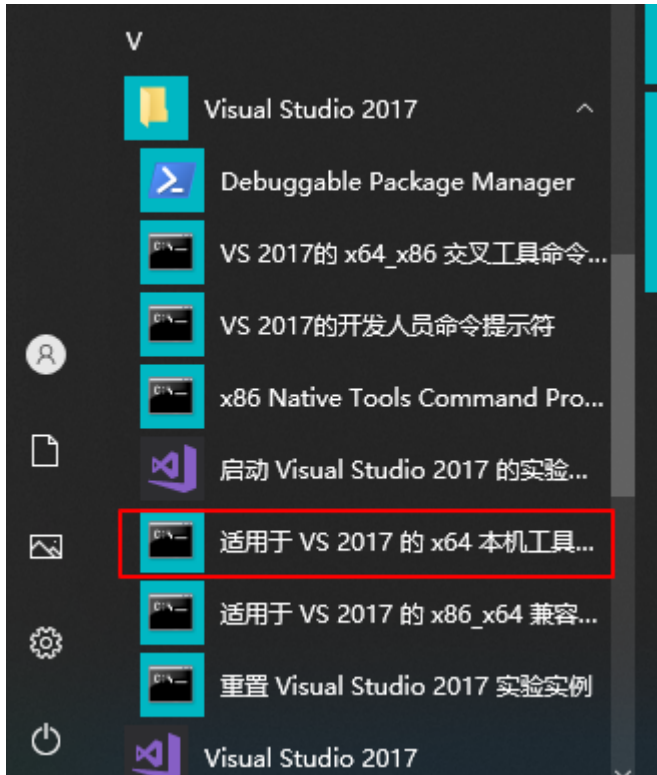
---

官网<https://slproweb.com/products/Win32OpenSSL.html>，下载后直接安装，并添加系统环境变量

```
OPENSSL_ROOT_DIR    C:\Program Files\OpenSSL-Win64
```

## 2.3 Memory-0.7-1

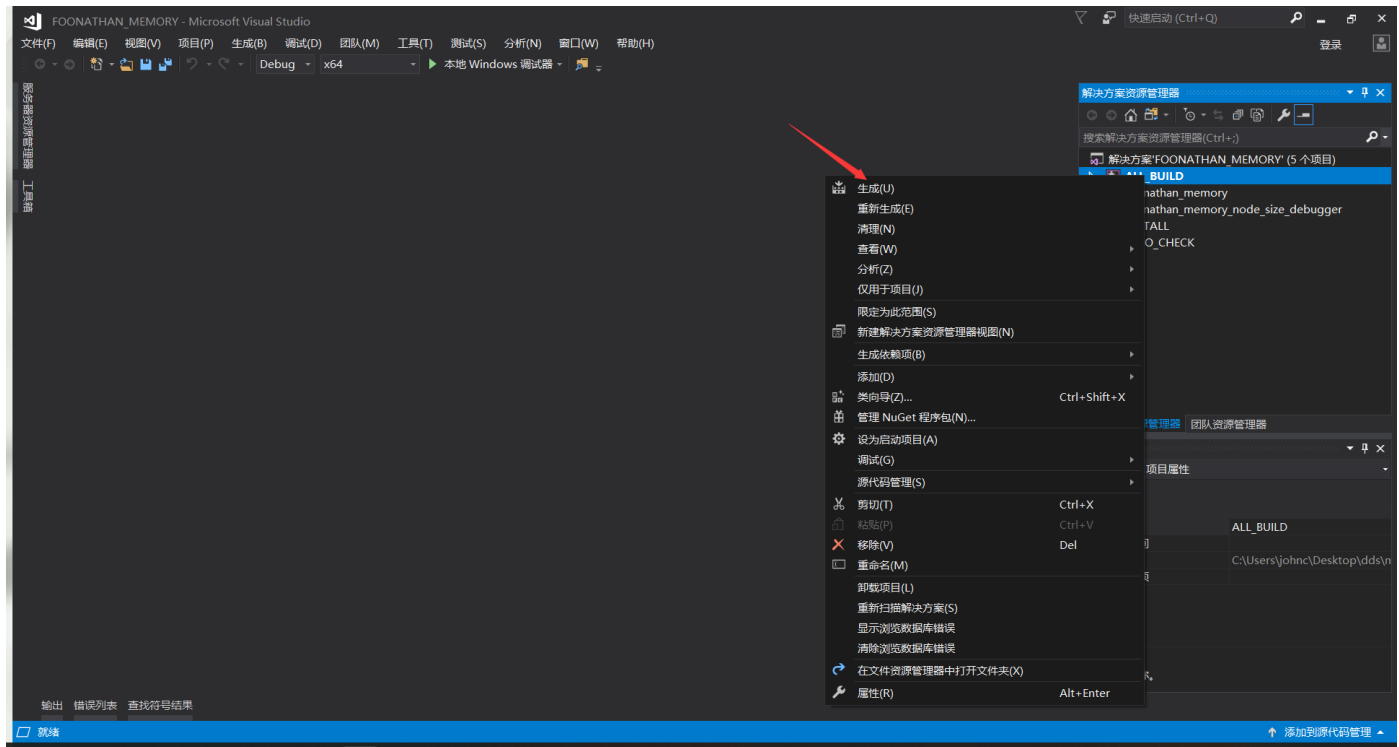
1、以管理员身份 打开vs 2017 command 并进入项目build目录



2、Build目录下运行

```
cmake -G "Visual Studio 15 2017 Win64" .. -DBUILD_SHARED_LIBS=false -  
DCMAKE_BUILD_TYPE="buildtype" -DFOONATHAN_MEMORY_BUILD_EXAMPLES=OFF -  
DFOONATHAN_MEMORY_BUILD_TESTS=OFF
```

没问题的话会在build下生成vs解决方案，点开.sln文件后选择build all生成



### 3、安装

```
cmake --build . --target install
```

默认install目录为C:\Program Files，也可自己指定目录。

## 2.4 tinyxml2

1、以管理员身份 打开vs 2017 command 并进入项目build目录，Build目录下运行：

```
cmake -G "Visual Studio 15 2017 Win64" .. -DBUILD_SHARED_LIBS=false
```

2、打开vs进行编译生成

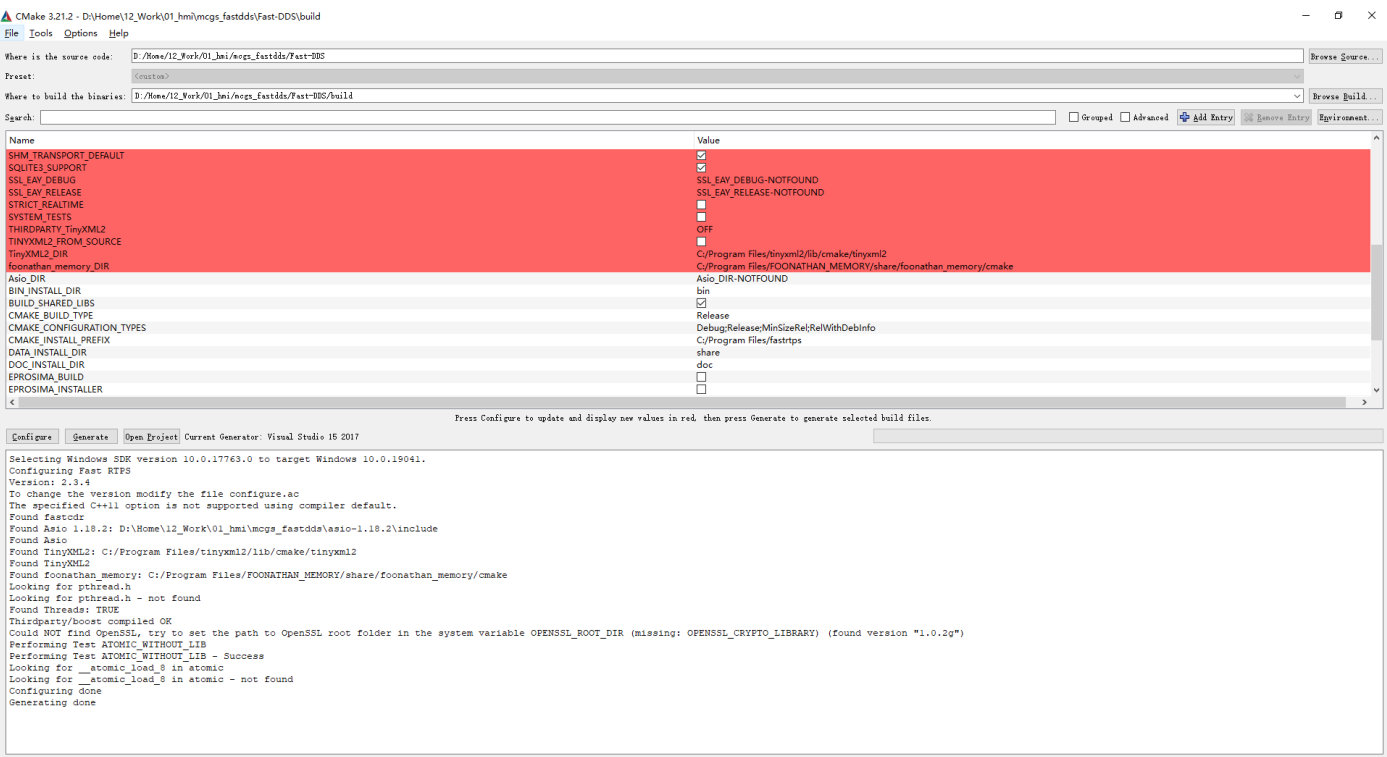
3、安装

```
cmake --build . --target install
```

## 2.5 Fast-CDR

## 2.6 Fast-DDS

1、进入build目录，打开cmake，可能根据需要配置asio或tinyxml路径，asio配置（即ASIO\_INCLUDE\_DIR那一行）其include目录即可，tinyxml配置如下图，如果弹窗错误，退出cmake重新打开几次即可



2、生成sln解决方案

3、打开vs进行编译生成

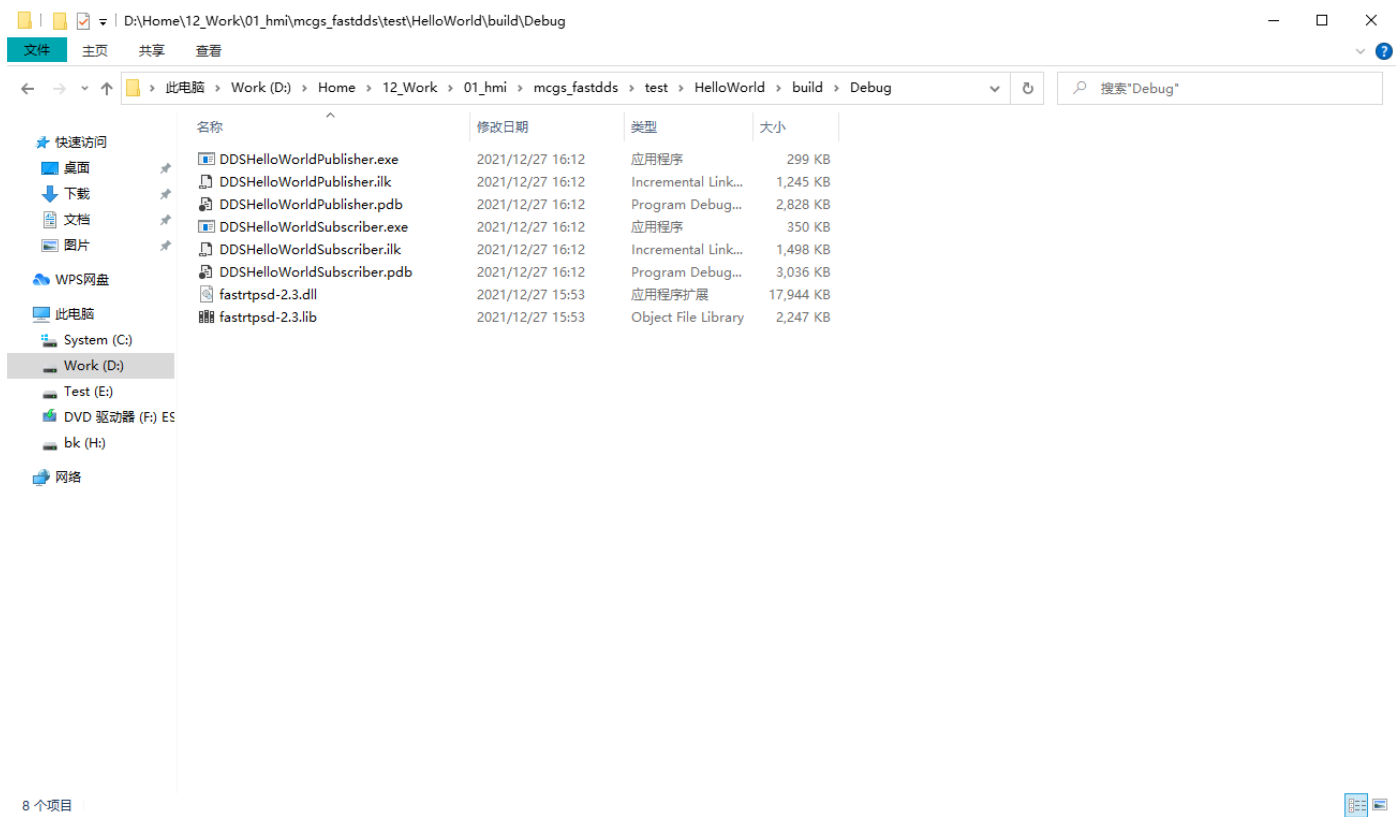
4、安装

```
cmake --build . --target install
```

## 3 测试DDS

使用Helloword例子进行验证

Helloword编译步骤同tinyxml2，最后一步无需install，生成DDSHelloWorldPublisherexe和DDSHelloWorldSubscriber.exe后，注意需要拷贝fastrtps-2.3.dll到该目录。



可以修改下图数值改变发送次数：

```
HelloWorldPublisher.cpp  X
DDSHelloWorldPublisher>HelloWorldPublisher
190 while (samples_sent < samples)
191 {
192     if (publish())
193     {
194         samples_sent++;
195         // std::cout << "Message: " << hello_.message() << " with index: " << hello_.index()
196         // << " SENT" << std::endl;
197         std::cout << " SENT:" << samples_sent << std::endl;
198     }
199     std::this_thread::sleep_for(std::chrono::milliseconds(50));
200 }
201 }
202 };
203
204 int main(
205     int argc,
206     char **argv)
207 {
208     std::cout << "Starting publisher." << std::endl;
209     int samples = 10//10000;
210
211     HelloWorldPublisher *mypub = new HelloWorldPublisher();
212     if (mypub->init())
213     {
214         mypub->run(static_cast<uint32_t>(samples));
215     }
216
217     delete mypub;
218     return 0;
219 }
220
```

```
PS D:\Home\12_Work\01_hmi\mcgs_fastdds\test\HelloWorld\build\Debug> .\DDSHelloWorldPublisher.exe
Starting publisher.
Publisher matched.
at addr:000000BA922FF998 SENT:1
at addr:000000BA922FF998 SENT:2
at addr:000000BA922FF998 SENT:3
at addr:000000BA922FF998 SENT:4
at addr:000000BA922FF998 SENT:5
at addr:000000BA922FF998 SENT:6
at addr:000000BA922FF998 SENT:7
at addr:000000BA922FF998 SENT:8
at addr:000000BA922FF998 SENT:9
at addr:000000BA922FF998 SENT:10
```

```
PS D:\Home\12_Work\01_hmi\mcgs_fastdds\test\HelloWorld\build\Debug> .\DDSHelloWorldSubscriber.exe
Recv addr at000000234B4450114 731699
Recv addr at000000234B4650194 2019500
Recv addr at000000234B4450114 1138300
Recv addr at000000234B4650194 2882899
Recv addr at000000234B4450114 1580700
Recv addr at000000234B4650194 2048199
Recv addr at000000234B4450114 851700
Recv addr at000000234B4650194 2109699
Recv addr at000000234B4450114 1389699
Recv addr at000000234B4650194 1808000
```