

# 1、实验名称及目的

**DDS 组网通信：**配置 DDS 组网需要的环境。自行搭建 DDS 协议并创建收发端口实现 DDS 通信。

# 2、实验原理

DDS 是一种实时系统的数据分发服务标准，它采用发布-订阅模式，通过中间的数据分发服务代理让发布者和订阅者无需知道彼此的存在就能交换信息。其核心原理包括灵活的数据模型、主题的定义与使用、内容过滤、QoS 策略、发现服务以及安全等。自建协议可能是指基于这些 DDS 原理，开发出一套符合自身应用需求的具体协议实现，涉及到定制化的数据模型、主题定义、QoS 策略及安全保障措施等。

# 3、实验效果

```
F:\work\Git\9_RflySimComm\SourceCode\DDS组网通信例程\windows\Fast-DDS-python-1.2.0\Demo>python HelloWorldExample.py -
p publisher
Received {message}
Creating Start.
Creating publisher.
Writer is waiting discovery...
Publisher matched subscriber 1.f.a7.60.60.69.24.f7.0.0.0.0.0.1.4
Writer discovery finished...
Sending Hello World : 0
Sending Hello World : 1
Sending Hello World : 2
Sending Hello World : 3
Sending Hello World : 4
Sending Hello World : 5
Sending Hello World : 6
Sending Hello World : 7
Sending Hello World : 8
Sending Hello World : 9

F:\work\Git\9_RflySimComm\SourceCode\DDS组网通信例程\windows\Fast-DDS-python-1.2.0\Demo>python HelloWorldExample.py -
p subscriber
Received {message}
Creating Start.
Creating subscriber.
Press any key to stopSubscriber matched publisher 1.f.a7.60.44.27.94.4b.0.0.0.0.0.1.3
Received Hello World : 0
Received Hello World : 1
Received Hello World : 2
Received Hello World : 3
Received Hello World : 4
Received Hello World : 5
Received Hello World : 6
Received Hello World : 7
Received Hello World : 8
Received Hello World : 9
Subscriber unmatched publisher 1.f.a7.60.44.27.94.4b.0.0.0.0.0.1.3
```

# 4、文件目录

文件夹/文件名称	说明
HolleWorld	通信格式
Demo	实验例程文件
fastdds_python	Fast DDS 提供的 Python 绑定库
fastdds_python_examples	Fast DDS 提供的示例

## 5、运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 <sup>①</sup>	1
2	RflySim 平台免费版		

## 6、实验步骤

### 一、Windows 系统从二进制文件安装 eProsima Fast DDS

#### 1.安装准备

Windows 环境下，从二进制文件安装 eProsima Fast DDS 需要安装以下工具:Visual Studio。Visual Studio 需要在系统中有一个 c++编译器。为此，确保在 Visual Studio 安装过程中选中了 Desktop development with c++选项。

如果已经安装了 Visual Studio，但是没有安装 Visual c++ Redistributable 包，打开 Visual Studio，然后转到 Tools -> 获取工具和功能，并在工作负载选项卡中启用使用 c++进行桌面开发。最后，单击右下角的 Modify。

#### 2.安装 DDS

最新版本的 eProsima Fast DDS Windows 可在公司网站下载页面 [Downloads \(eprosima.com\)](https://www.eprosima.com/downloads)。下载完成后，执行安装程序并按照说明执行，在提示时选择首选的 Visual Studio 版本和体系结构。

#### 3.内容

默认情况下，安装会下载所有可用的软件包，即：

foonathan\_memory\_vendor: 一个 STL 兼容的 c++内存分配器库。

fastcdr: 一个 c++库，根据标准 CDR 序列化机制进行序列化。

fasttps: eProsima Fast DDS 库的核心库。

fastddsgen: 一个使用 IDL 文件中定义的数据类型生成源代码的 Java 应用程序。

#### 4.系统环境变量

eProsima Fast DDS 需要以下环境变量设置才能正常工作：

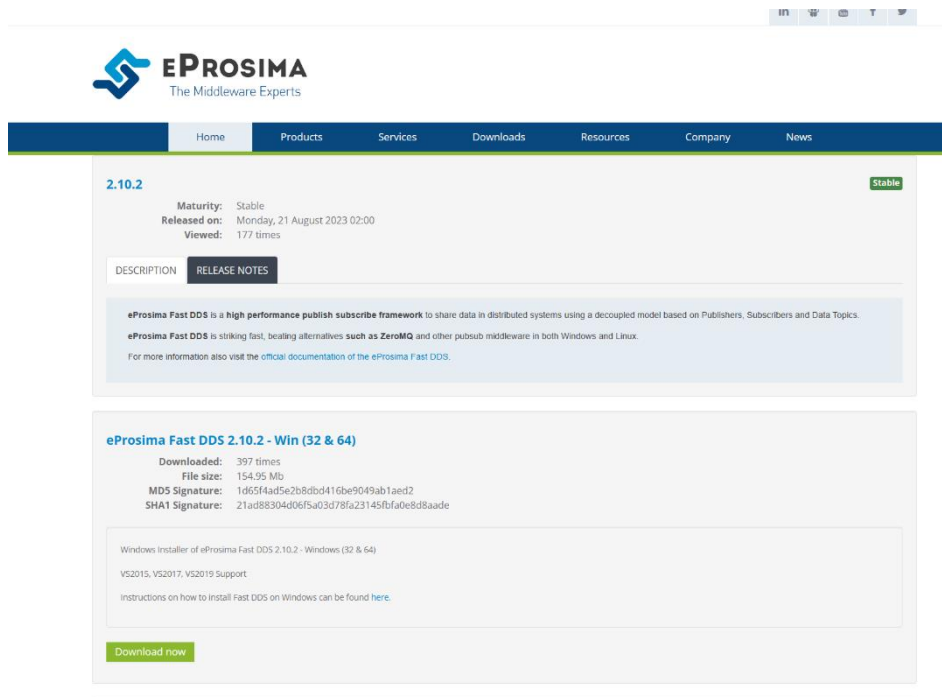
FASTRTPSHOME:安装 eProsima Fast DDS 的根文件夹。

添加到 PATH: eProsima Fast DDS 脚本和库的位置应该添加到 PATH。

这些变量在安装过程中通过勾选相应的复选框自动设置。

#### 5.软件下载安装：

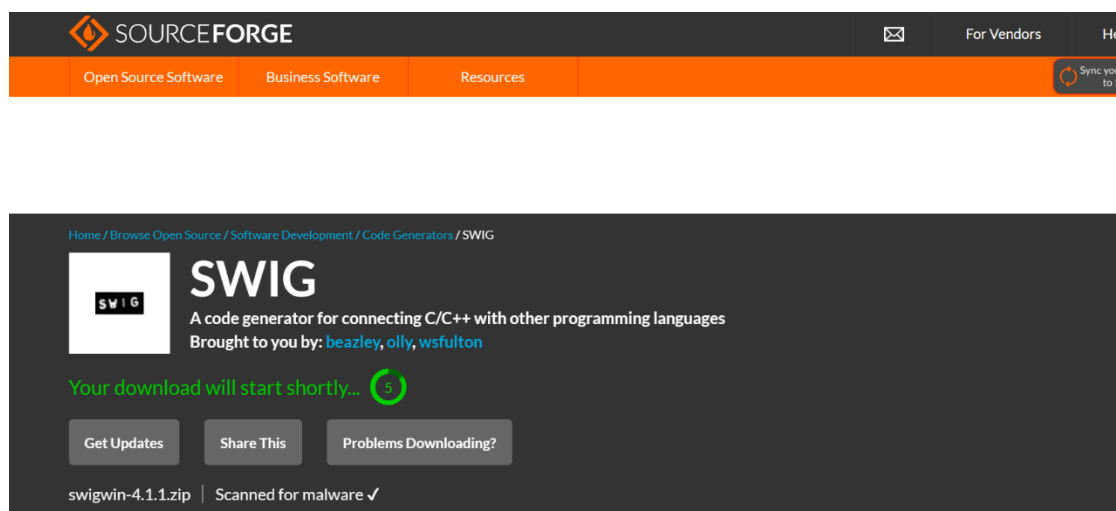
Windows 系统从源安装：[eProsima Fast DDS 2.10.2](https://www.eprosima.com/downloads)

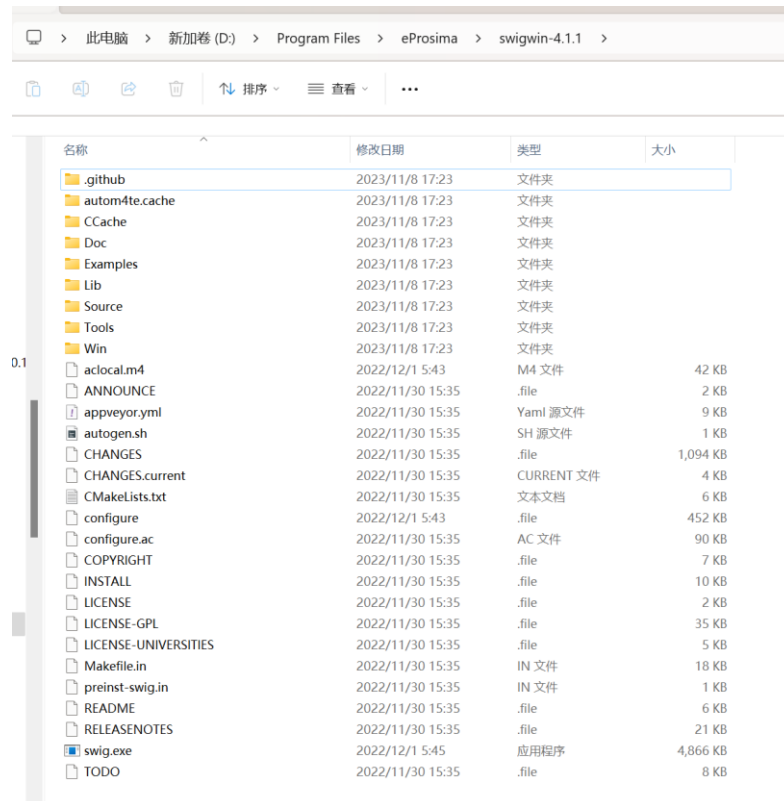


## 二、编译安装 fastdds\_python

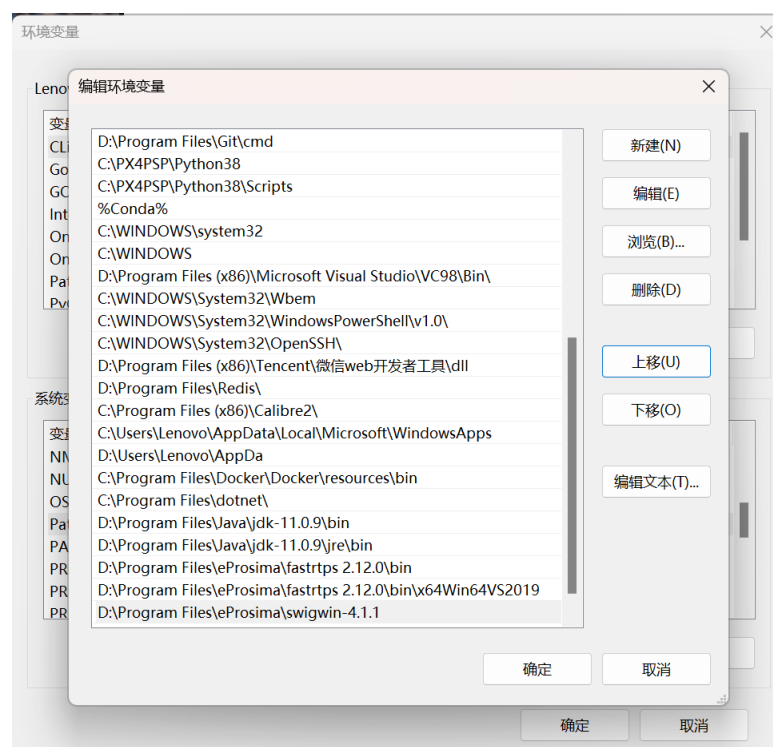
1、由于编译 fastdds\_python 需要用到 SWIG, 此处需要安装 SWIG(解压至相应文件夹), 我这里安装的 SWIG 版本为 4.1.1 (这里需注意 swig 版本需要与 fastdds 版本对应, 这里 Fastdds 版本是 1.10.2)。

下载链接: [Download swigwin-4.1.1.zip \(SWIG\) \(sourceforge.net\)](https://sourceforge.net/projects/swig/files/swigwin-4.1.1.zip/download)



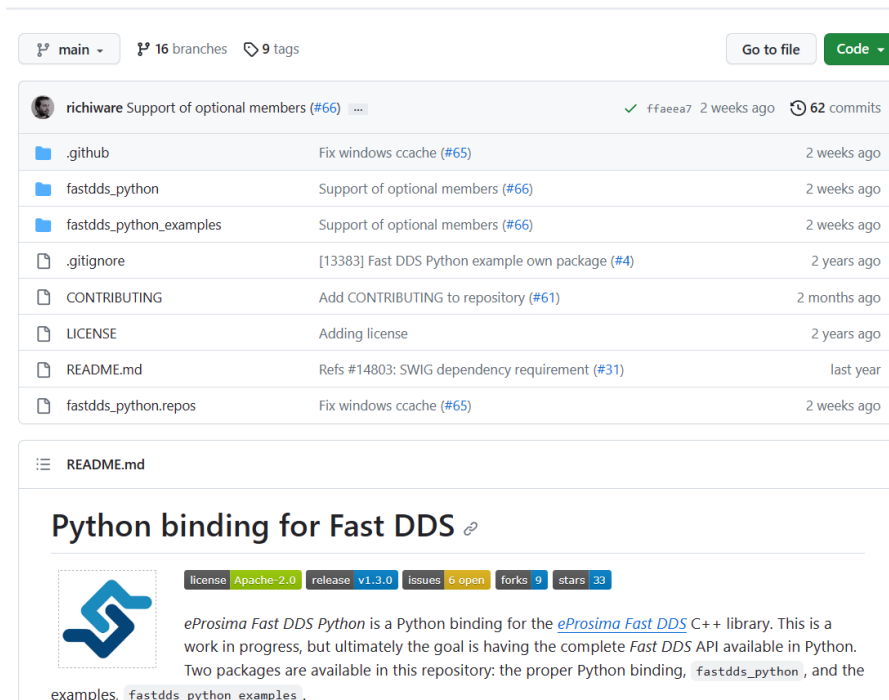


需要配置到环境变量

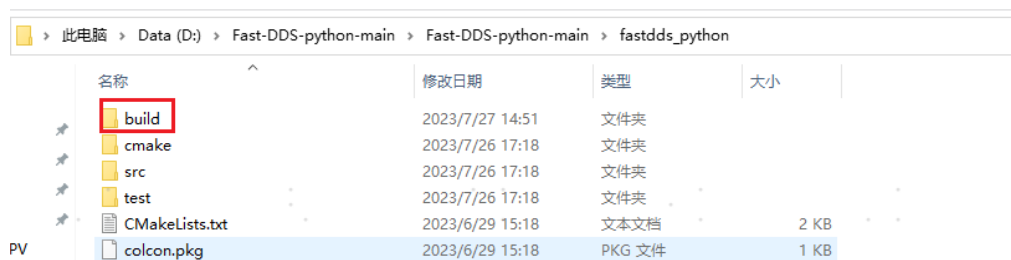


2、从 github 上下载 fastdds\_python

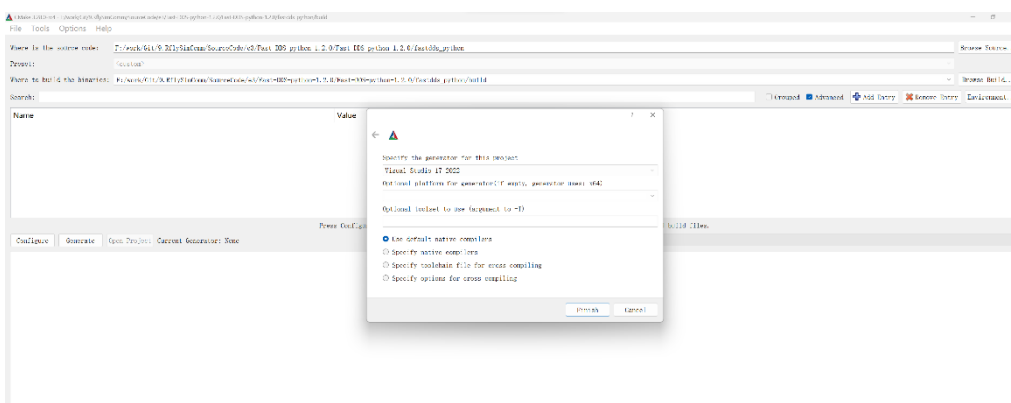
下载链接: <https://github.com/eProsima/Fast-DDS-python>

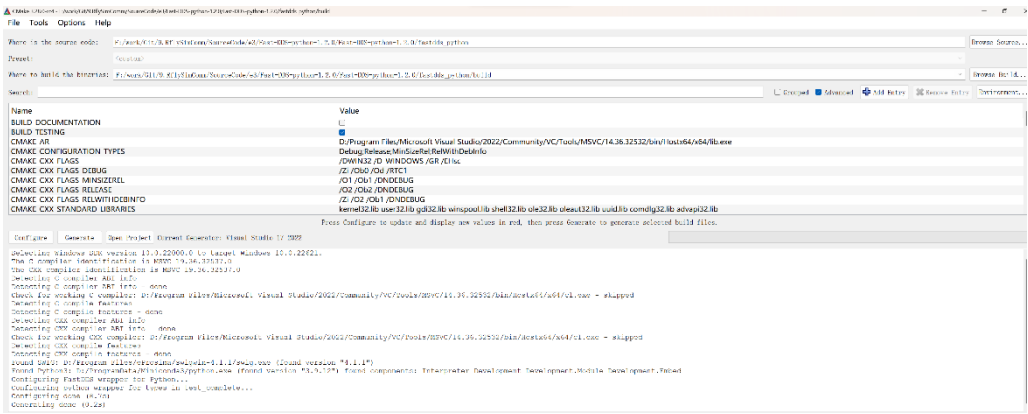


3、内容包括两部分 fastdds 库和一个例子，我们首先编译 fastdds\_python。首先在 fastdds\_python 目录下新建文件夹 build。

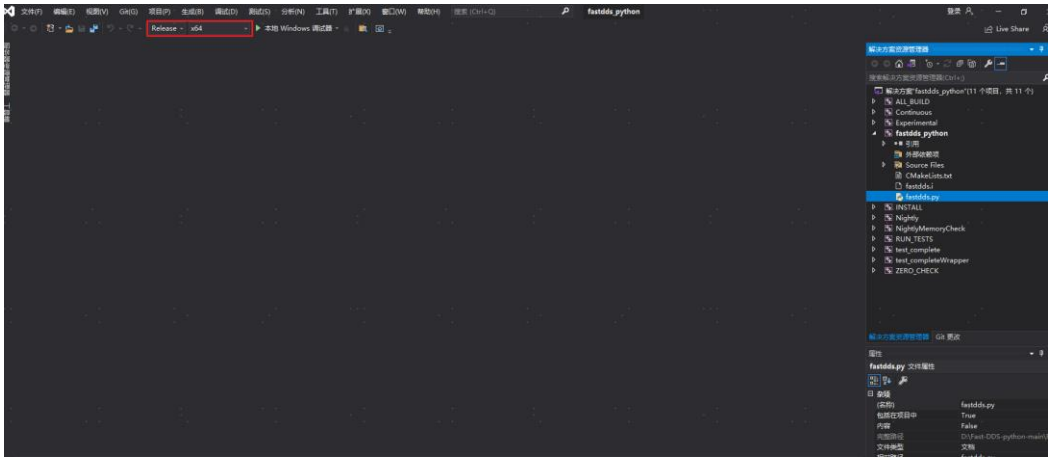


在这里我们使用 cmake-gui 工具构建工程，具体工程请参考《CMake-基础介绍》CSDN，点击 Generate，就会出现配置界面配置完后点 finish，然后就会自动编译，可能会出现 Openssl include 文件夹找不到或者 swig 执行文件找不到，可以手动配置。最后编译完成后点击 Open Project，则会自动使用 Visual Studio 打开项目。

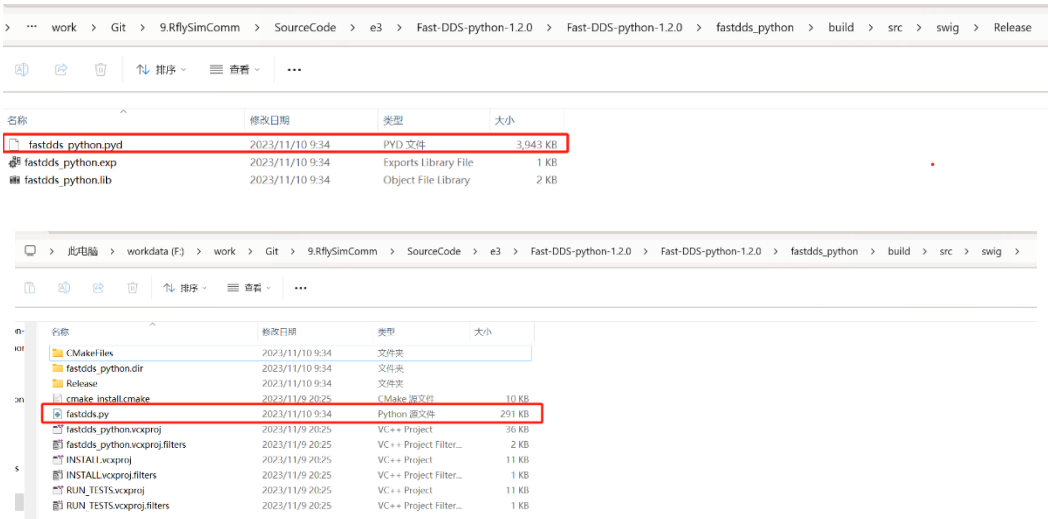




将编译模式转成 Release，然后在解决方案资源管理器中右键选择全部重新生成。



编译成功后，fastdds.py 和 \_fastdds\_python.pyd 这两个文件用于后面 python 工程。



### 三、DDS 通信示例：

下面我们新建一个文件夹 HelloWorld，并在里面新建 IDL 文件 HelloWorld.idl，用于定义通信格式

```
struct HelloWorld
{
```

```

unsigned long index;
string message;
};

```

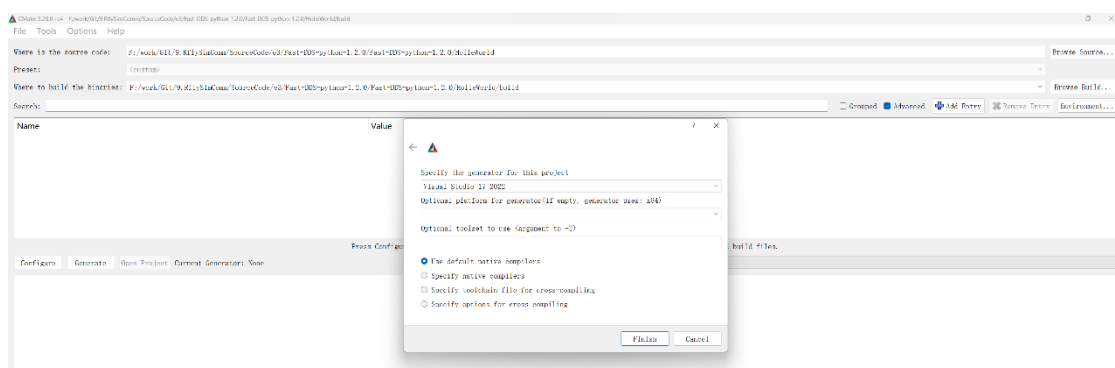
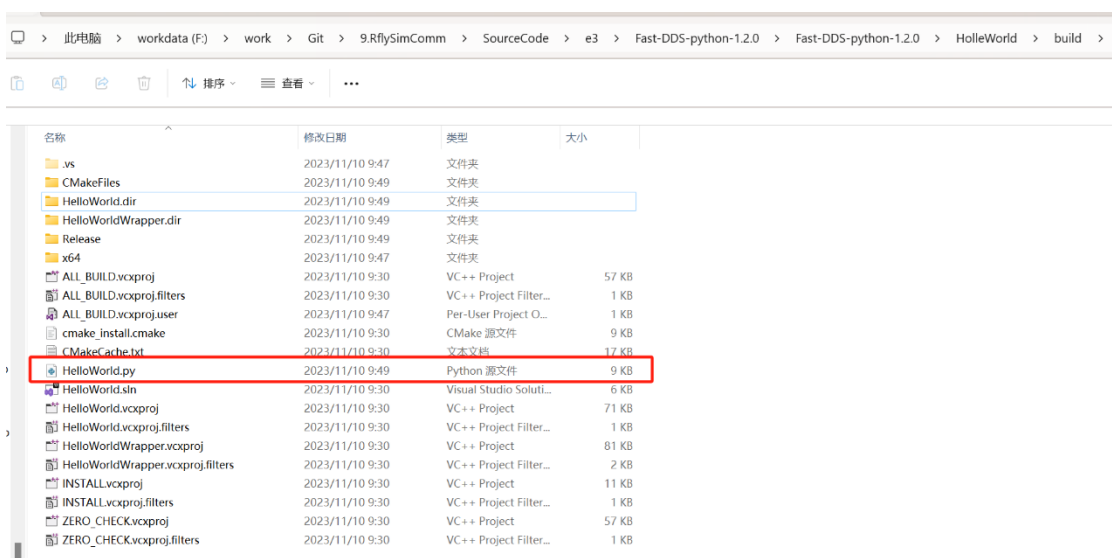
下面是 c++ 头文件的包含。第一个包含 HelloWorldPubSubTypes.h 文件，其中包含在新建的文件夹 HelloWorld 中，打开 cmd，并在 cmd 中使用 fastddsgen.bat 命令生成 python 使用的通信协议文件

fastddsgen.bat -python HelloWorld.idl -ppDisable  
得到内容如下：



名称	修改日期	类型	大小
build	2023/11/9 20:24	文件夹	
CMakeLists.txt	2023/11/9 18:50	文本文件	4 KB
HelloWorld.cxx	2023/11/9 18:50	QtProject.QtCrea...	5 KB
HelloWorld.h	2023/11/9 18:50	C Header 源文件	7 KB
HelloWorld.i	2023/11/9 18:50	Preprocessed C/C+...	4 KB
HelloWorld.idl	2023/11/9 14:26	Interface Definition ...	1 KB
HelloWorldPubSubTypes.cxx	2023/11/9 18:50	QtProject.QtCrea...	6 KB
HelloWorldPubSubTypes.h	2023/11/9 18:50	C Header 源文件	4 KB
HelloWorldPubSubTypes.i	2023/11/9 18:50	Preprocessed C/C+...	1 KB

重复上述编译操作的过程，打开 VS2022 最终生成的文件，后面需要用的文件有 \_HelloWorldWrapper.pyd、HelloWorld.dll 和 HelloWorld.py

名称	修改日期	类型	大小
vs	2023/11/10 9:47	文件夹	
CMakeFiles	2023/11/10 9:49	文件夹	
HelloWorld.dir	2023/11/10 9:49	文件夹	
HelloWorldWrapper.dir	2023/11/10 9:49	文件夹	
Release	2023/11/10 9:49	文件夹	
x64	2023/11/10 9:47	文件夹	
ALL_BUILD.vcxproj	2023/11/10 9:30	VC++ Project	57 KB
ALL_BUILD.vcxproj.filters	2023/11/10 9:30	VC++ Project Filter...	1 KB
ALL_BUILD.vcxproj.user	2023/11/10 9:47	Per-User Project O...	1 KB
cmake_install.cmake	2023/11/10 9:30	CMake 源文件	9 KB
CMakeCache.txt	2023/11/10 9:30	文本文件	17 KB
HelloWorld.py	2023/11/10 9:49	Python 源文件	9 KB
HelloWorld.sln	2023/11/10 9:30	Visual Studio Solu...	6 KB
HelloWorld.vcxproj	2023/11/10 9:30	VC++ Project	71 KB
HelloWorld.vcxproj.filters	2023/11/10 9:30	VC++ Project Filter...	1 KB
HelloWorldWrapper.vcxproj	2023/11/10 9:30	VC++ Project	81 KB
HelloWorldWrapper.vcxproj.filters	2023/11/10 9:30	VC++ Project Filter...	2 KB
INSTALL.vcxproj	2023/11/10 9:30	VC++ Project	11 KB
INSTALL.vcxproj.filters	2023/11/10 9:30	VC++ Project Filter...	1 KB
ZERO_CHECK.vcxproj	2023/11/10 9:30	VC++ Project	57 KB
ZERO_CHECK.vcxproj.filters	2023/11/10 9:30	VC++ Project Filter...	1 KB

此电脑 > workdata (F:) > work > Git > 9.RflySimComm > SourceCode > e3 > Fast-DDS-python-1.2.0 > Fast-DDS-python-1.2.0 > HolleWorld > build > Release

名称	修改日期	类型	大小
_HelloWorldWrapper.pyd	2023/11/10 9:49	PYD 文件	129 KB
HelloWorld.dll	2023/11/10 9:49	应用程序扩展	41 KB
HelloWorld.exp	2023/11/10 9:49	Exports Library File	9 KB
HelloWorld.lib	2023/11/10 9:49	Object File Library	14 KB
HelloWorldWrapper.exp	2023/11/10 9:49	Exports Library File	1 KB
HelloWorldWrapper.lib	2023/11/10 9:49	Object File Library	2 KB

然后新建一个文件夹 Demo，把 fastdds\_python\_example 自带例子中的 HelloWorldExample.py 和之前生成的文件拷贝至同一个文件夹 Demo 内。得到如下图内容：

此电脑 > workdata (F:) > work > Git > 9.RflySimComm > SourceCode > e3 > Fast-DDS-python-1.2.0 > Fast-DDS-python-1.2.0 > Demo >

名称	修改日期	类型	大小
_pycache_	2023/11/9 19:40	文件夹	
_fastdds_python.pyd	2023/11/9 19:35	PYD 文件	3,942 KB
_HelloWorldWrapper.pyd	2023/11/9 19:39	PYD 文件	129 KB
fastdds.py	2023/11/9 19:35	Python 源文件	291 KB
HelloWorld.dll	2023/11/9 19:39	应用程序扩展	41 KB
HelloWorld.py	2023/11/9 19:39	Python 源文件	9 KB
HelloWorldExample.py	2023/11/9 19:48	Python 源文件	8 KB

最后在当前路径下打开两个 cmd 终端，分别执行

```
python HelloWorldExample.py -p publisher
python HelloWorldExample.py -p subscriber
```

得到如图所示效果（可能需要多试几次）：

```
F:\work\Git\9.RflySimComm\SourceCode\e3\DDS组网通信例程\windows\Fast-DDS-python-1.2.0\Demo>python HelloWorldExample.py -p publisher
Received {message}
Creating Start.
Creating publisher.
Writer is waiting discovery...
Publisher matched subscriber 1.f.a7.60.60.69.24.f7.0.0.0.0.0.1.4
Writer discovery finished...
Sending Hello World : 0
Sending Hello World : 1
Sending Hello World : 2
Sending Hello World : 3
Sending Hello World : 4
Sending Hello World : 5
Sending Hello World : 6
Sending Hello World : 7
Sending Hello World : 8
Sending Hello World : 9
```

```
F:\work\Git\9.RflySimComm\SourceCode\e3\DDS组网通信例程\windows\Fast-DDS-python-1.2.0\Demo>python HelloWorldExample.py -p subscriber
Received {message}
Creating Start.
Creating subscriber.
Press any key to stopSubscriber matched publisher 1.f.a7.60.44.27.94.4b.0.0.0.0.0.1.3
Received Hello World : 0
Received Hello World : 1
Received Hello World : 2
Received Hello World : 3
Received Hello World : 4
Received Hello World : 5
Received Hello World : 6
Received Hello World : 7
Received Hello World : 8
Received Hello World : 9
Subscriber unmatched publisher 1.f.a7.60.44.27.94.4b.0.0.0.0.0.1.3
```

## 7、参考资料

## 8、常见问题