

DDS组网通信

1. 实验目的

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. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]。
- 硬件要求：笔记本/台式电脑1台^[2]。

3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\9.RflySimComm\0.ApiExps\e2.DDSDeptsTestExps](#)

- [./HolleWorld](#)：通信格式相关文件。
- [./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。

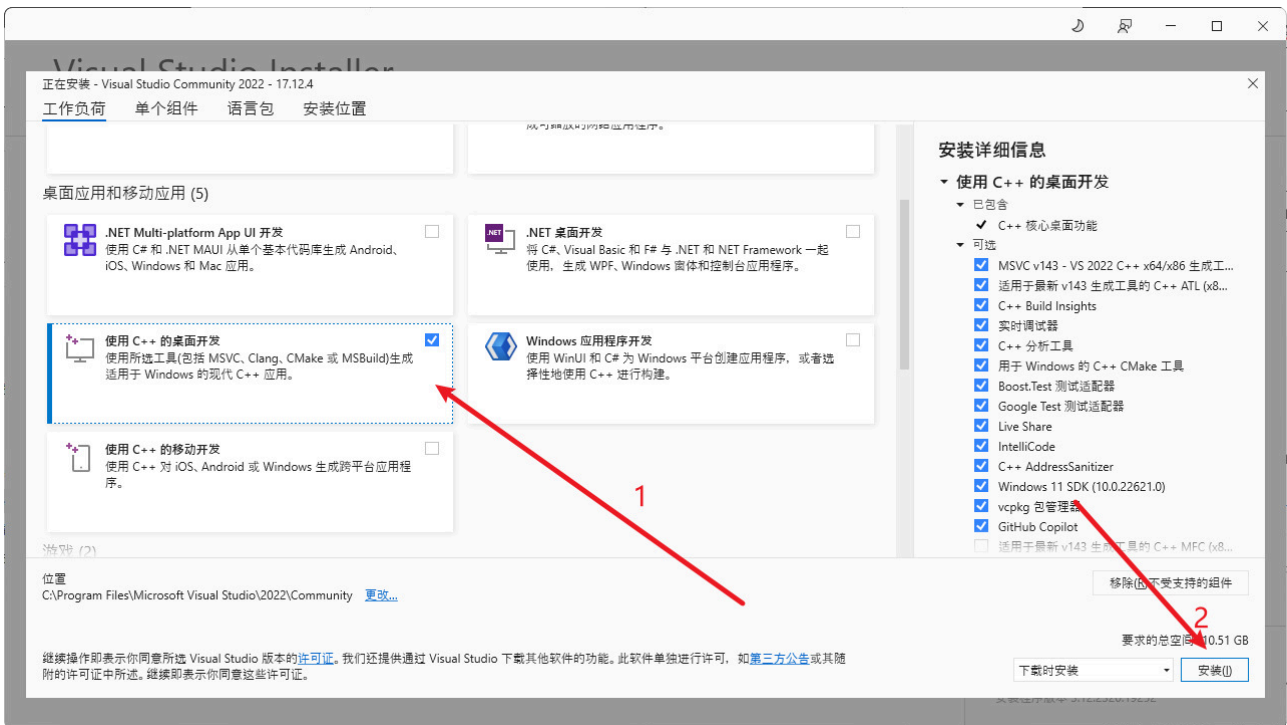
4. 实验内容或步骤

4.1 步骤1：Windows系统从二进制文件安装eProsima Fast DDS

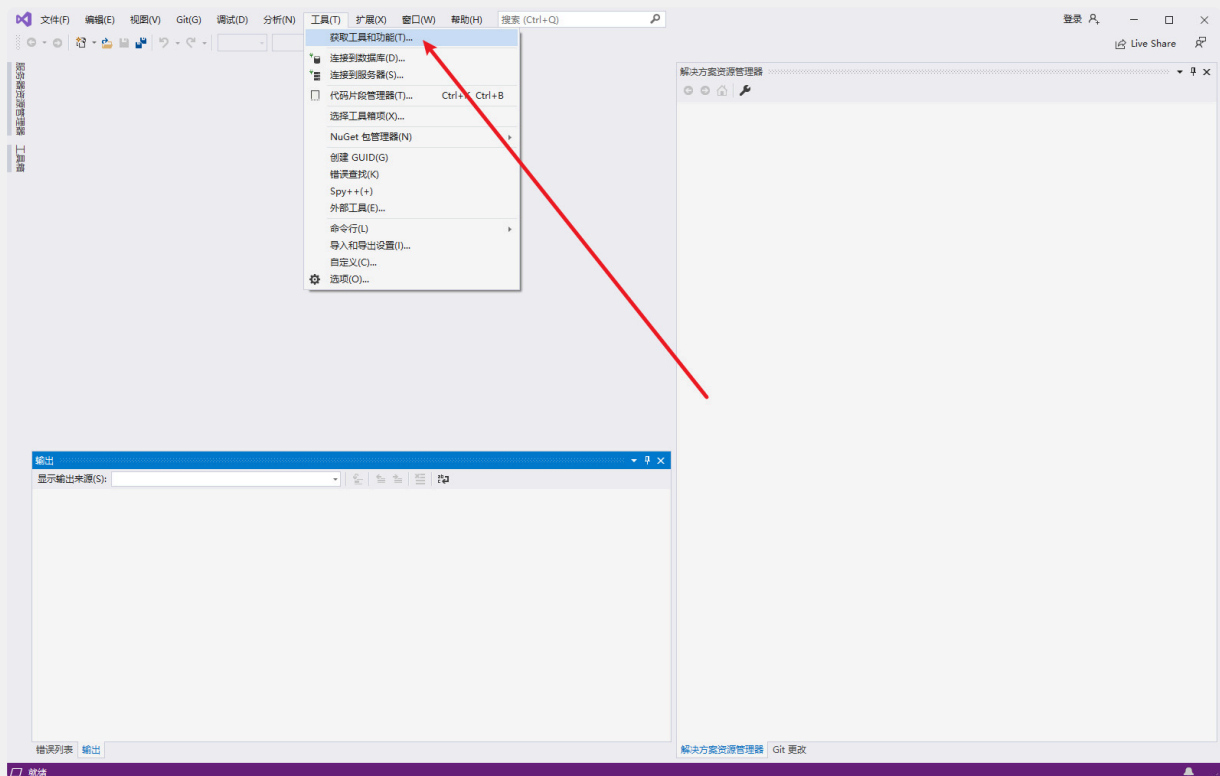
1. 安装准备

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。

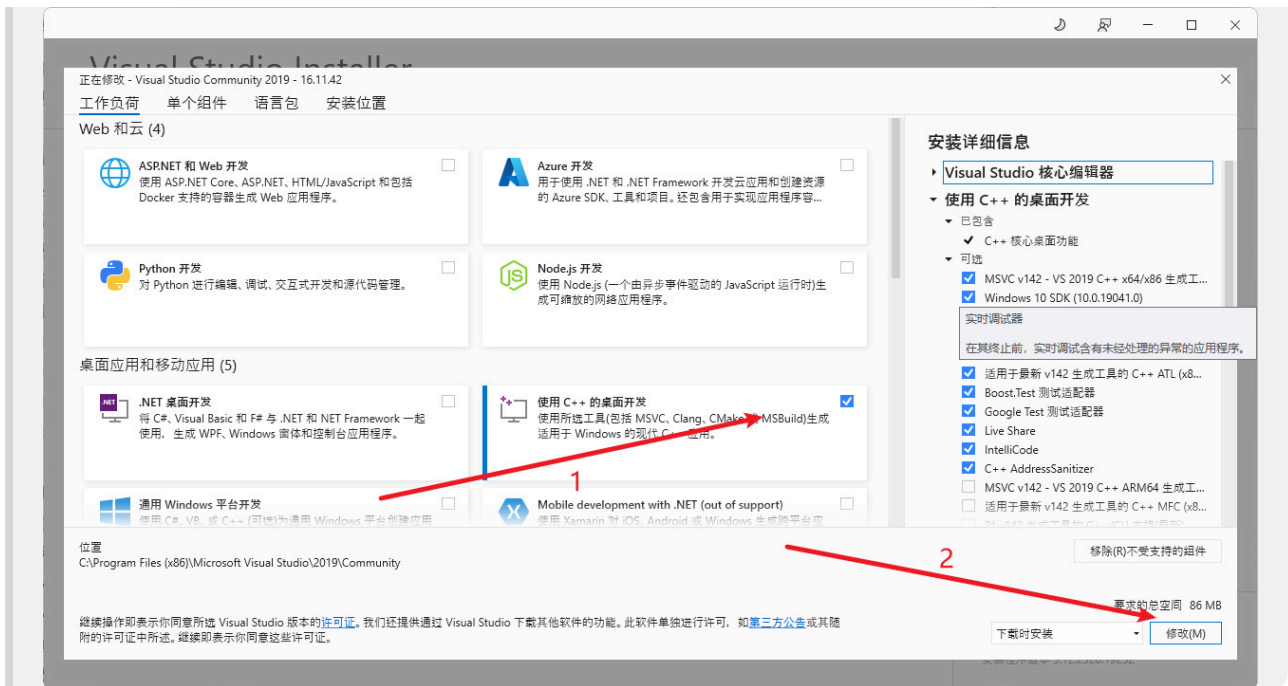
从 Visual Studio: 面向软件开发人员和 Teams 的 IDE 和代码编辑器 下载 Visual Studio 安装包，双击打开安装包；在如下界面仅勾选"使用C++的桌面开发"，最后点击安装。



如果已经安装了 Visual Studio，但是没有安装 Visual C++ Redistributable 包，打开 Visual Studio，然后点击"工具"下的"获取工具和功能"按钮。



并在弹出的选项卡中启用"使用C++进行桌面开发"。最后，单击右下角的修改等待完成安装。



2. 安装CMake-GUI:

从官网 [Download CMake](#) 下载CMake安装程序，按照步骤执行安装。

Latest Release (3.31.4)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the linux-x86_64 tar file is all under the directory cmake-linux-x86_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.rst at the top of the source tree. See also the [CMake 3.31 Release Notes](#).

Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.31.4.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.31.4.zip

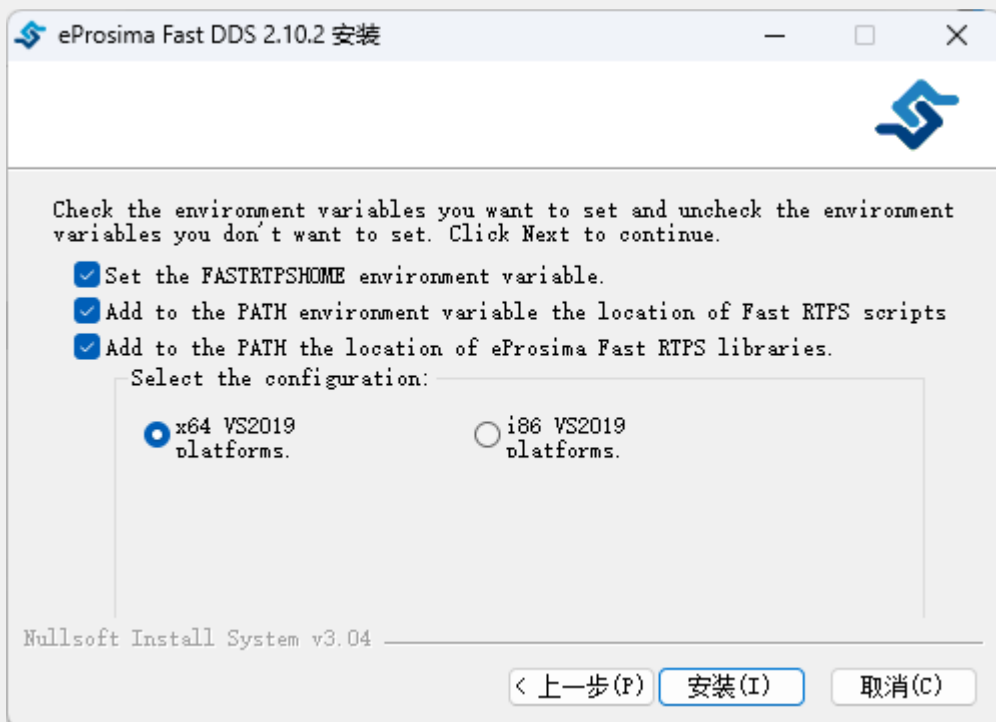
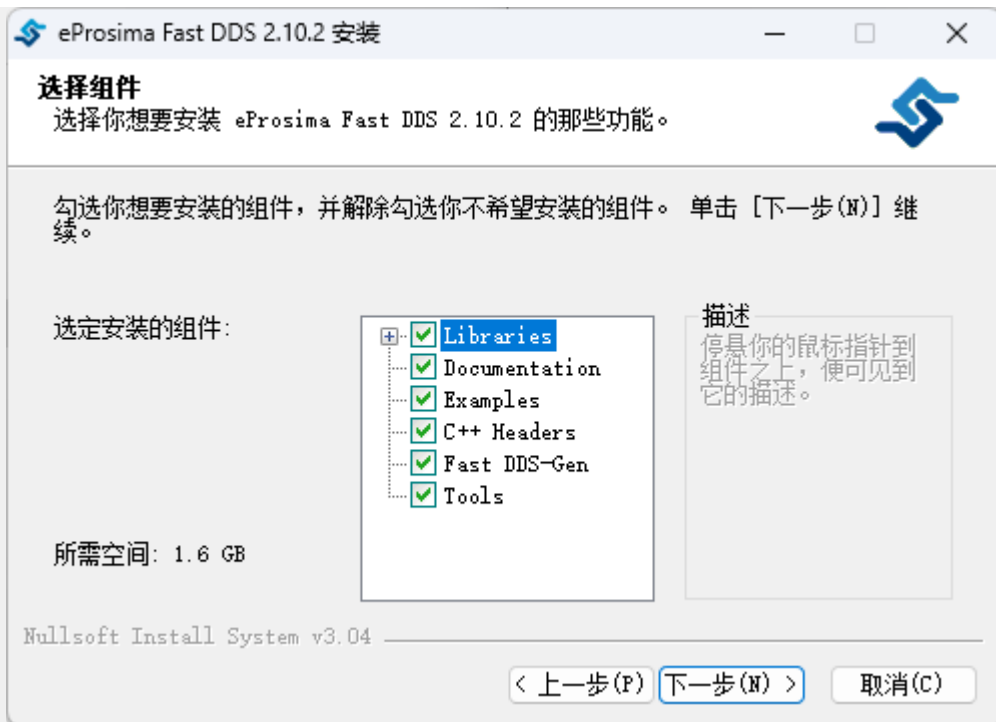
Binary distributions:

Platform	Files
Windows x64 Installer:	cmake-3.31.4-windows-x86_64.msi
Windows x64 ZIP	cmake-3.31.4-windows-x86_64.zip
Windows i386 Installer:	cmake-3.31.4-windows-i386.msi
Windows i386 ZIP	cmake-3.31.4-windows-i386.zip
Windows ARM64 Installer:	cmake-3.31.4-windows-arm64.msi
Windows ARM64 ZIP	cmake-3.31.4-windows-arm64.zip
macOS 10.13 or later	cmake-3.31.4-macos-universal.dmg
	cmake-3.31.4-macos-universal.tar.gz
macOS 10.10 or later	cmake-3.31.4-macos10.10-universal.dmg
	cmake-3.31.4-macos10.10-universal.tar.gz
Linux x86_64	cmake-3.31.4-linux-x86_64.sh
	cmake-3.31.4-linux-x86_64.tar.gz
Linux aarch64	cmake-3.31.4-linux-aarch64.sh
	cmake-3.31.4-linux-aarch64.tar.gz

3. 安装DDS

最新版本的eProxima Fast DDS Windows可在公司网站下载页面

[Downloads \(eprosima.com\)](#)。此处使用9.RflySimComm\0.ApiExps\1.DDS文件夹中的eProxima_Fast-DDS-2.10.2-Windows.exe安装包进行安装。执行安装程序并按照说明执行。



此处根据安装准备中Visual Studio版本和电脑体系结构选择合适的选项。

4. 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。

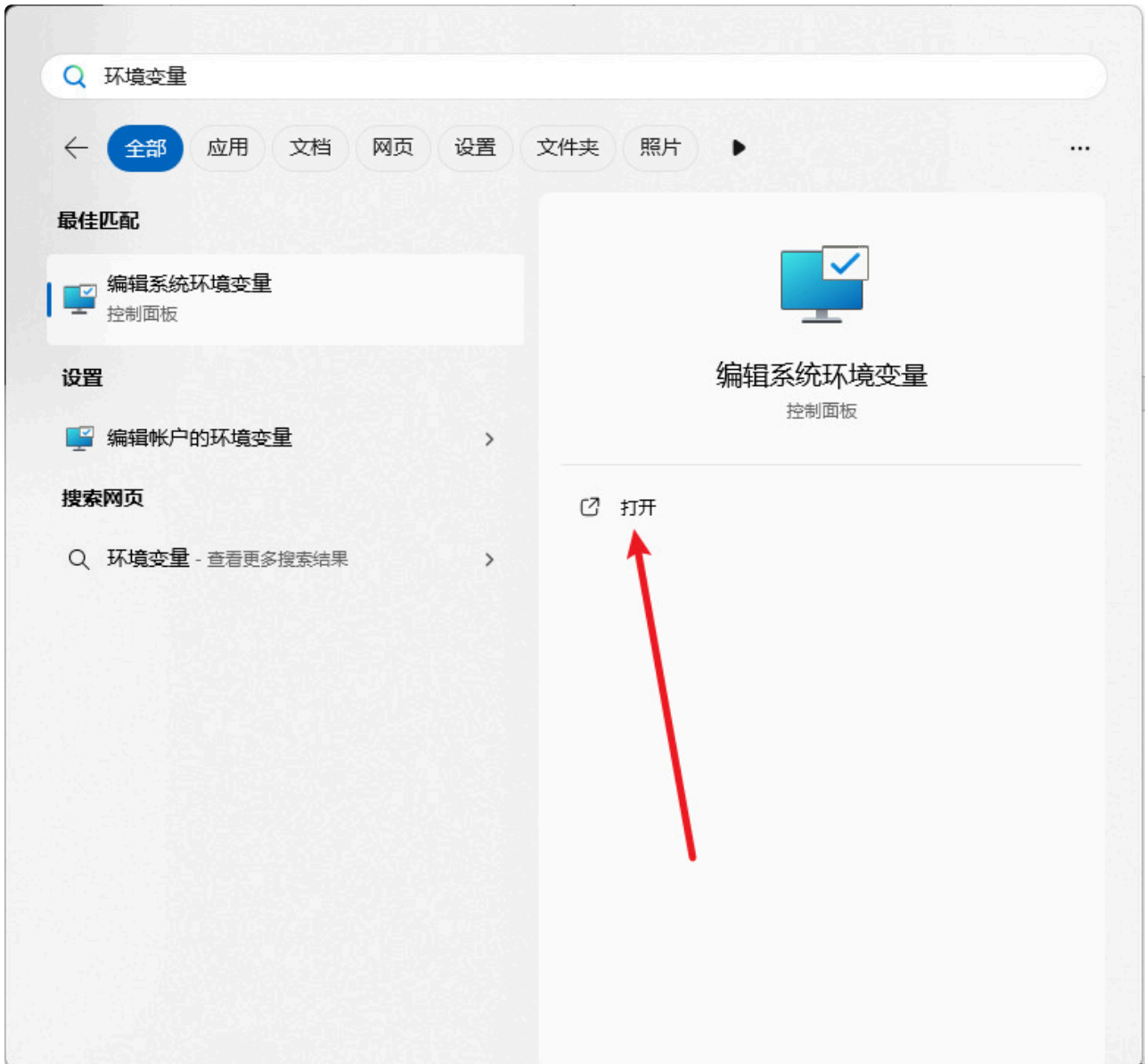
fasttps：eProsima Fast DDS库的核心库。

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。

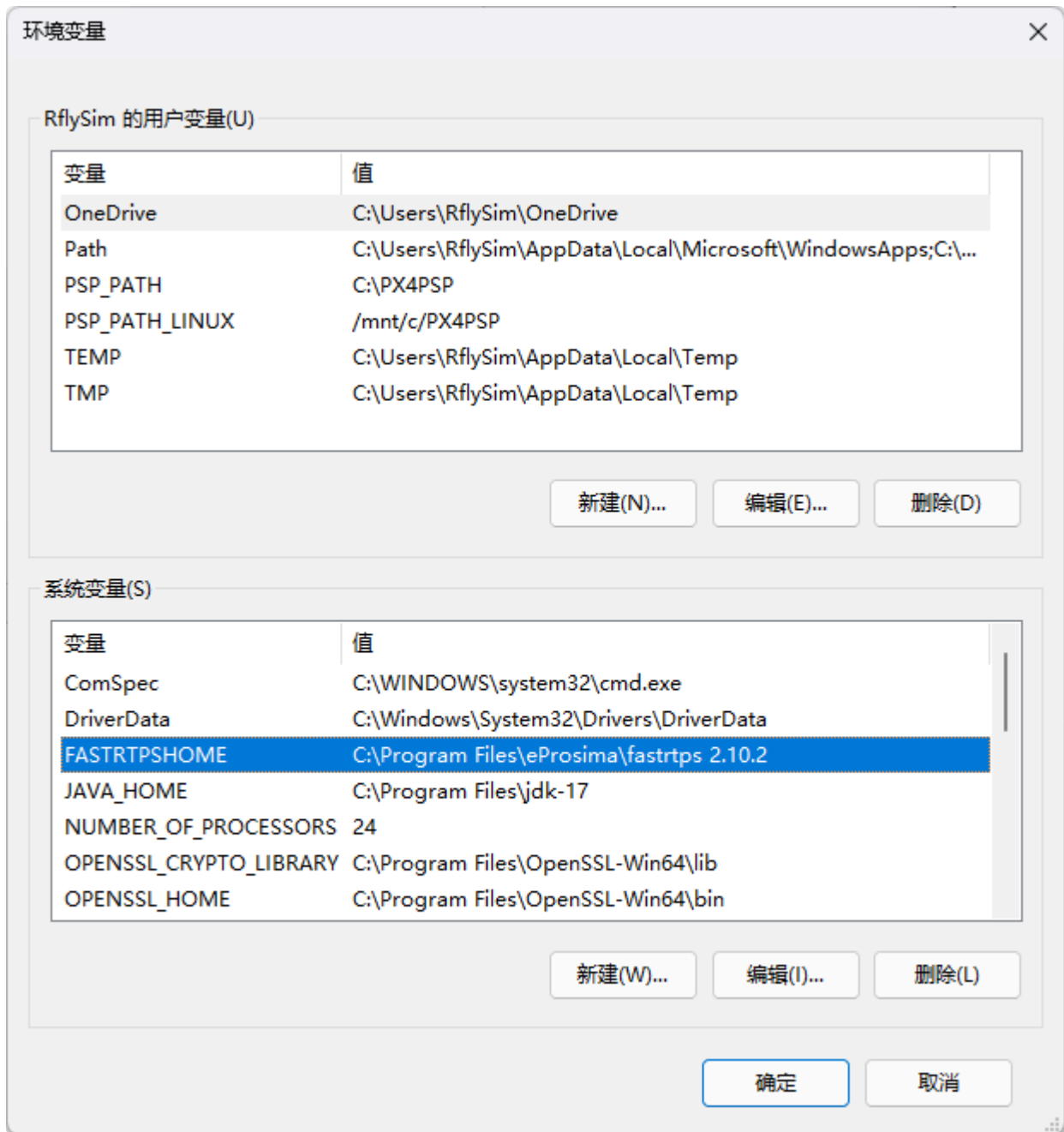
5. 系统环境变量

eProxima Fast DDS需要以下环境变量设置才能正常工作，安装程序会自动设置环境变量，此处仅为确认环境变量的正确设置：

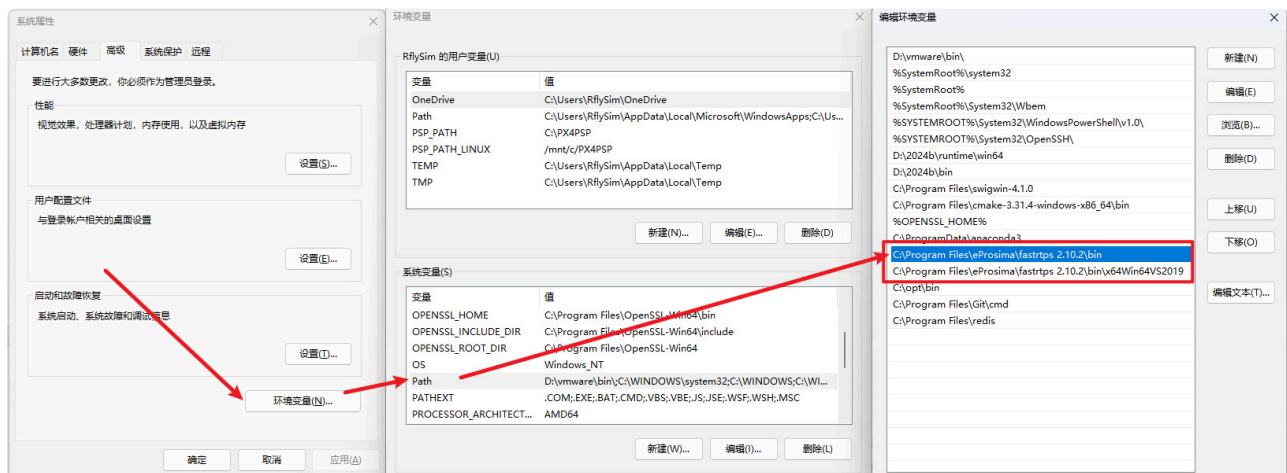
windows 11中在开始菜单旁的搜索栏中搜索"环境变量"：



FASTRTPSHOME:安装eProxima Fast DDS的根文件夹：



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

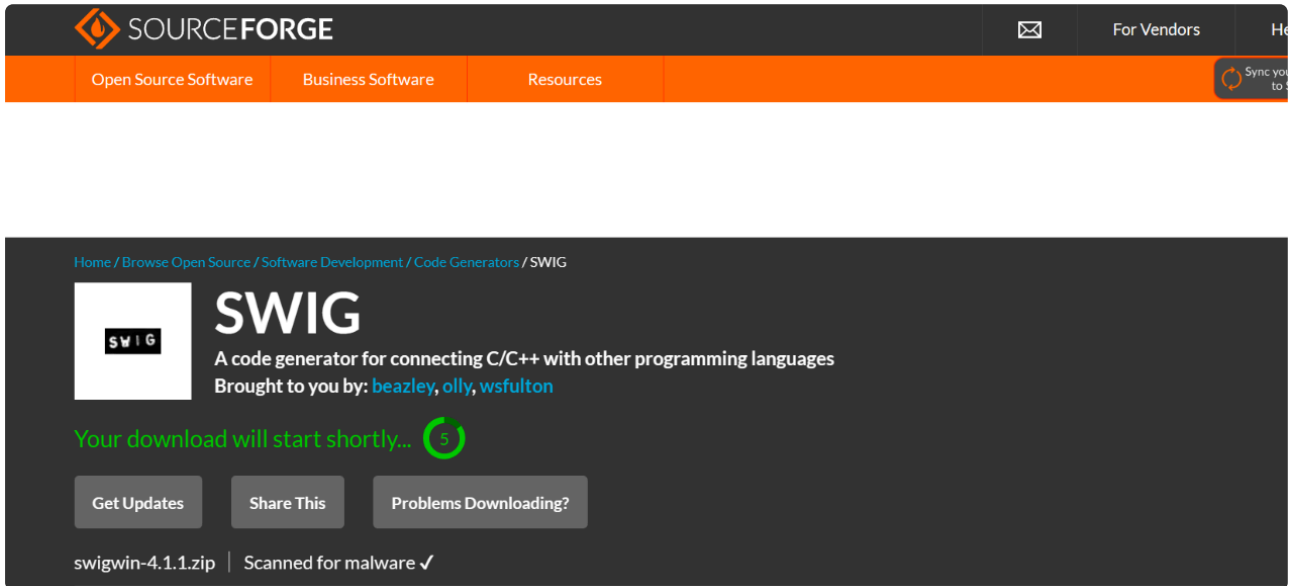


4.2 步骤2：安装 SWIG和OpenSSL

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）。

步骤1：下载SWIG：

下载链接：[Download swigwin-4.1.1.zip \(SWIG\) \(sourceforge.net\)](#)



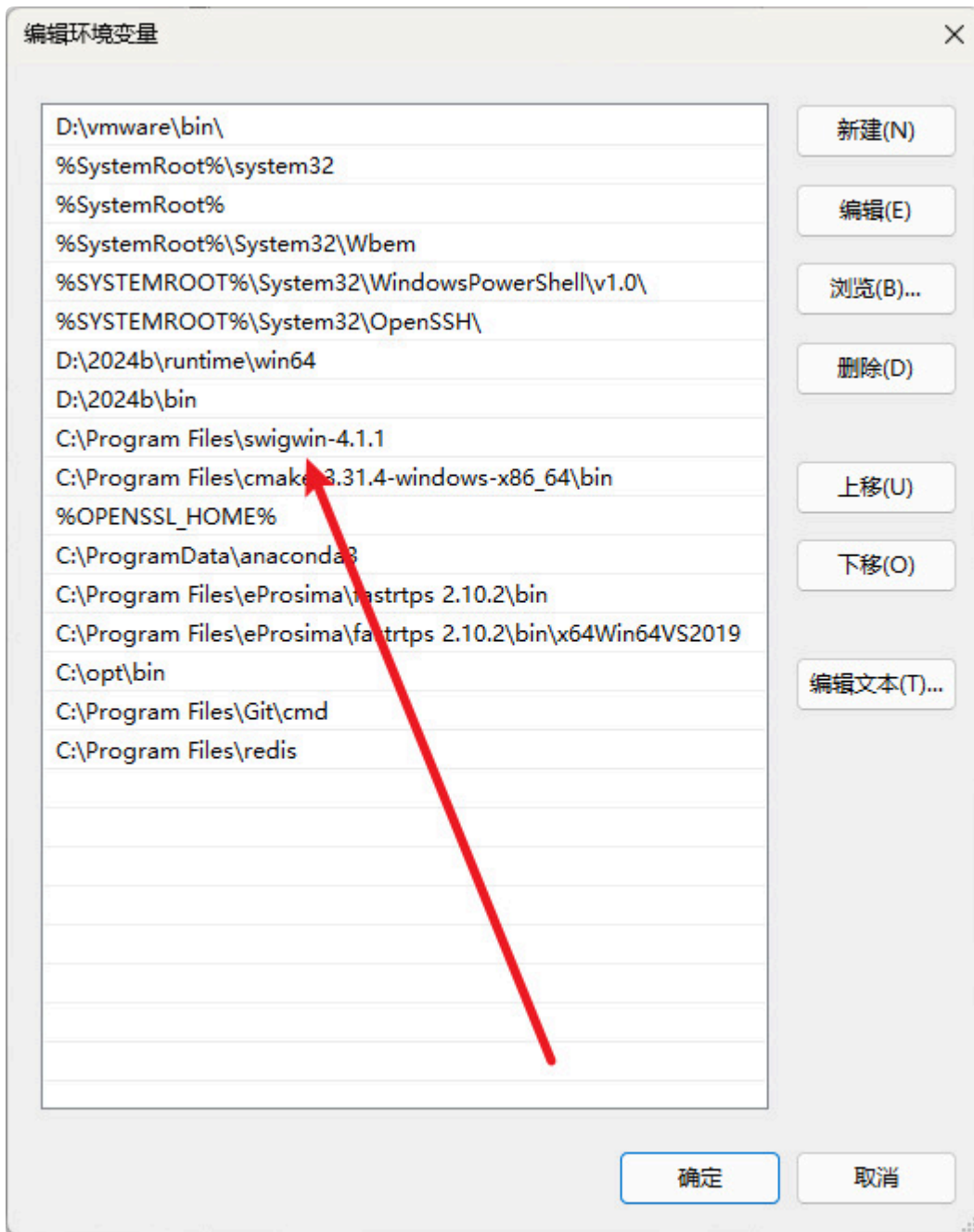
步骤2：解压安装SWIG：将swigwin-4.1.1.zip解压到指定目录，这里解压到C:\Program Files\swigwin-4.1.1中。

此电脑 > 本地磁盘 (C:) > Program Files > swigwin-4.1.1 >

排序 查看 ...

名称	修改日期	类型	大小
.github	2022/10/25 6:03	文件夹	
autom4te.cache	2022/10/25 6:40	文件夹	
CCache	2022/10/25 6:41	文件夹	
Doc	2022/10/25 6:03	文件夹	
Examples	2022/10/25 6:03	文件夹	
Lib	2022/10/25 6:40	文件夹	
Source	2022/10/25 6:40	文件夹	
Tools	2022/10/25 6:03	文件夹	
Win	2022/10/25 6:03	文件夹	
aclocal.m4	2022/10/25 6:40	M4 文件	42 KB
ANNOUNCE	2022/10/25 6:03	文件	2 KB
appveyor.yml	2022/10/25 6:03	Yaml 源文件	9 KB
autogen.sh	2022/10/25 6:03	sh_auto_file	1 KB
CHANGES	2022/10/25 6:03	文件	1,043 KB
CHANGES.current	2022/10/25 6:03	CURRENT 文件	52 KB
CMakeLists.txt	2022/10/25 6:03	文本文档	6 KB
configure	2022/10/25 6:40	文件	452 KB
configure.ac	2022/10/25 6:03	AC 文件	90 KB
COPYRIGHT	2022/10/25 6:03	文件	7 KB
INSTALL	2022/10/25 6:03	文件	10 KB
LICENSE	2022/10/25 6:03	文件	2 KB
LICENSE-GPL	2022/10/25 6:03	文件	35 KB
LICENSE-UNIVERSITIES	2022/10/25 6:03	文件	5 KB
Makefile.in	2022/10/25 6:03	IN 文件	18 KB
preinst-swig.in	2022/10/25 6:03	IN 文件	1 KB
README	2022/10/25 6:03	文件	6 KB
RELEASENOTES	2022/10/25 6:03	文件	21 KB
swig.exe	2022/10/25 6:41	应用程序	4,869 KB
TODO	2022/10/25 6:03	文件	8 KB

步骤3: 配置SWIG环境变量:



至此，SWIG的安装就完成了，下一步需要编译fastdds_python。

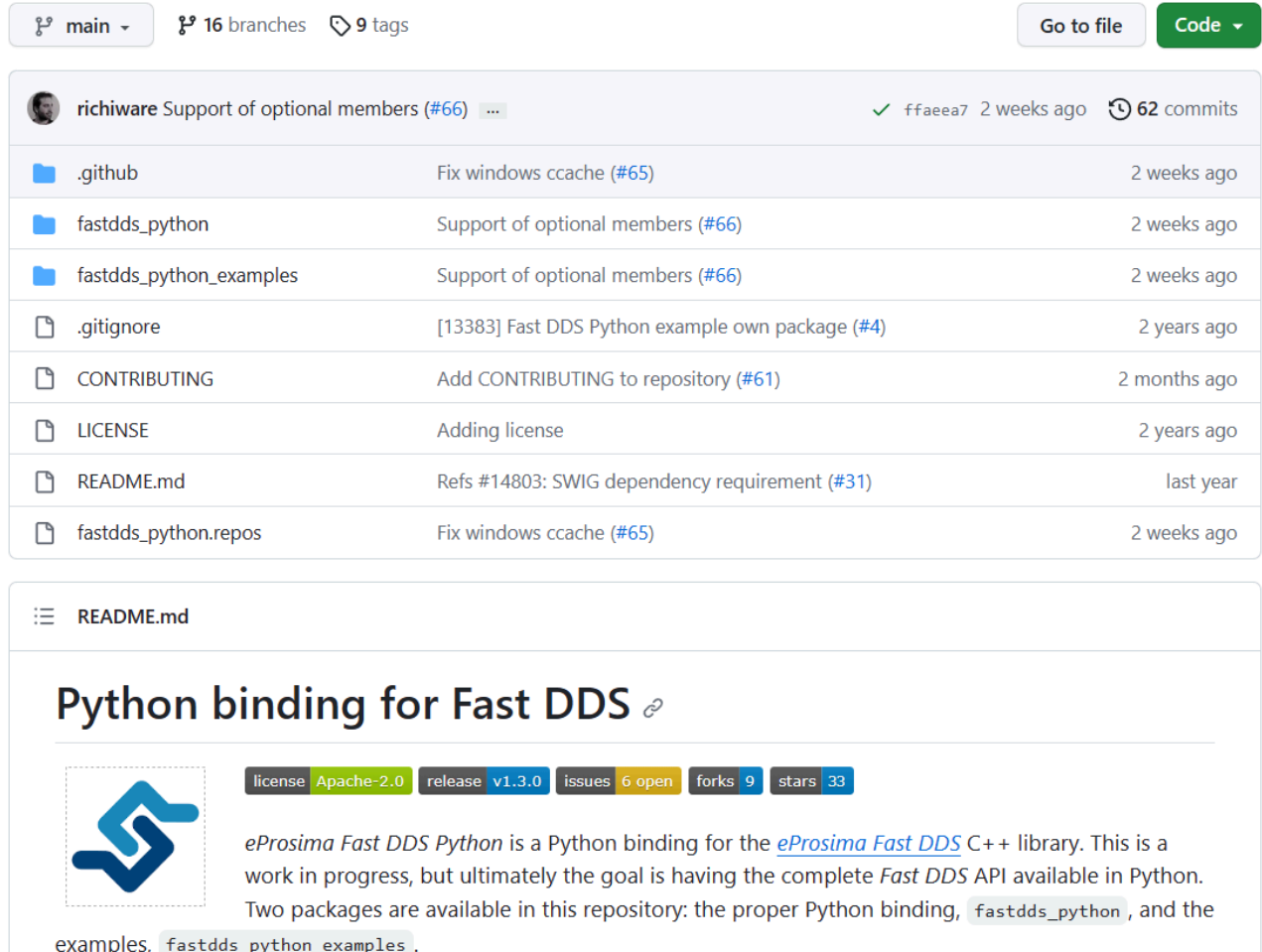
2. 由于编译fastdds_python需要用到OpenSSL，此处需要安装OpenSSL，从官网 [Win32/Win64 OpenSSL Installer for Windows - Shining Light Productions](https://www.openssl.org/source/openssl-winsource.exe) 下载安装包（EXE文件），直接双击安装：

File	Type	Description
Win64 OpenSSL v3.4.0 Light EXE MSI	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.4.0 (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.4.0 EXE MSI	221MB Installer	Installs Win64 OpenSSL v3.4.0 (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.4.0 Light EXE MSI	4MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v3.4.0 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.4.0 EXE MSI	180MB Installer	Installs Win32 OpenSSL v3.4.0 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.4.0 Light for ARM (EXPERIMENTAL) EXE MSI	6MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.4.0 for ARM64 devices (Only install this VERY EXPERIMENTAL build if you want to try 64-bit OpenSSL for Windows on ARM processors). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.4.0 for ARM (EXPERIMENTAL) EXE MSI	176MB Installer	Installs Win64 OpenSSL v3.4.0 for ARM64 devices (Only install this VERY EXPERIMENTAL build if you want to try 64-bit OpenSSL for Windows on ARM processors). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

4.3 步骤3：编译安装 fastdds_python

从github上下载fastdds_python

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



richiware Support of optional members (#66) ... ffaee7 2 weeks ago 62 commits

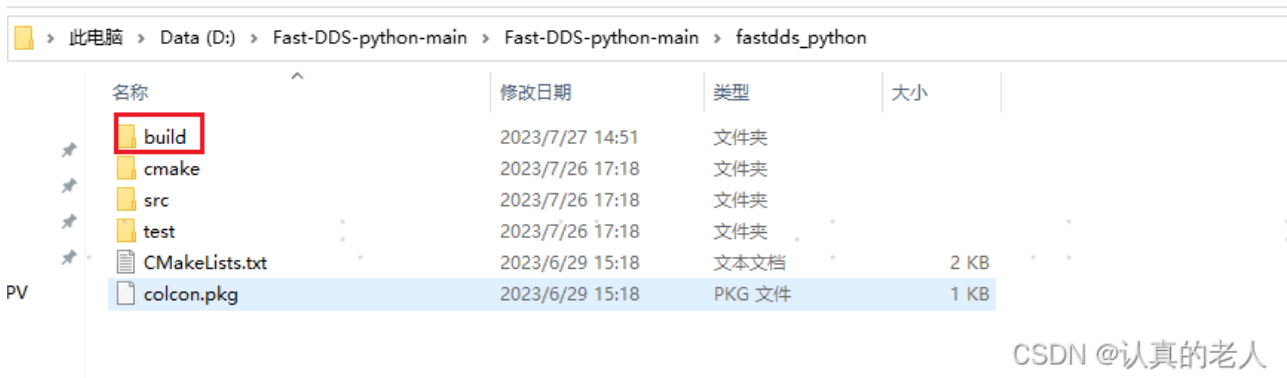
File	Commit	Time
.github	Fix windows ccache (#65)	2 weeks ago
fastdds_python	Support of optional members (#66)	2 weeks ago
fastdds_python_examples	Support of optional members (#66)	2 weeks ago
.gitignore	[13383] Fast DDS Python example own package (#4)	2 years ago
CONTRIBUTING	Add CONTRIBUTING to repository (#61)	2 months ago
LICENSE	Adding license	2 years ago
README.md	Refs #14803: SWIG dependency requirement (#31)	last year
fastdds_python.repos	Fix windows ccache (#65)	2 weeks ago

Python binding for Fast DDS

license Apache-2.0 release v1.3.0 issues 6 open forks 9 stars 33

eProsima Fast DDS Python is a Python binding for the eProsima Fast DDS C++ library. This is a work in progress, but ultimately the goal is having the complete Fast DDS API available in Python. Two packages are available in this repository: the proper Python binding, `fastdds_python`, and the examples, `fastdds_python_examples`.

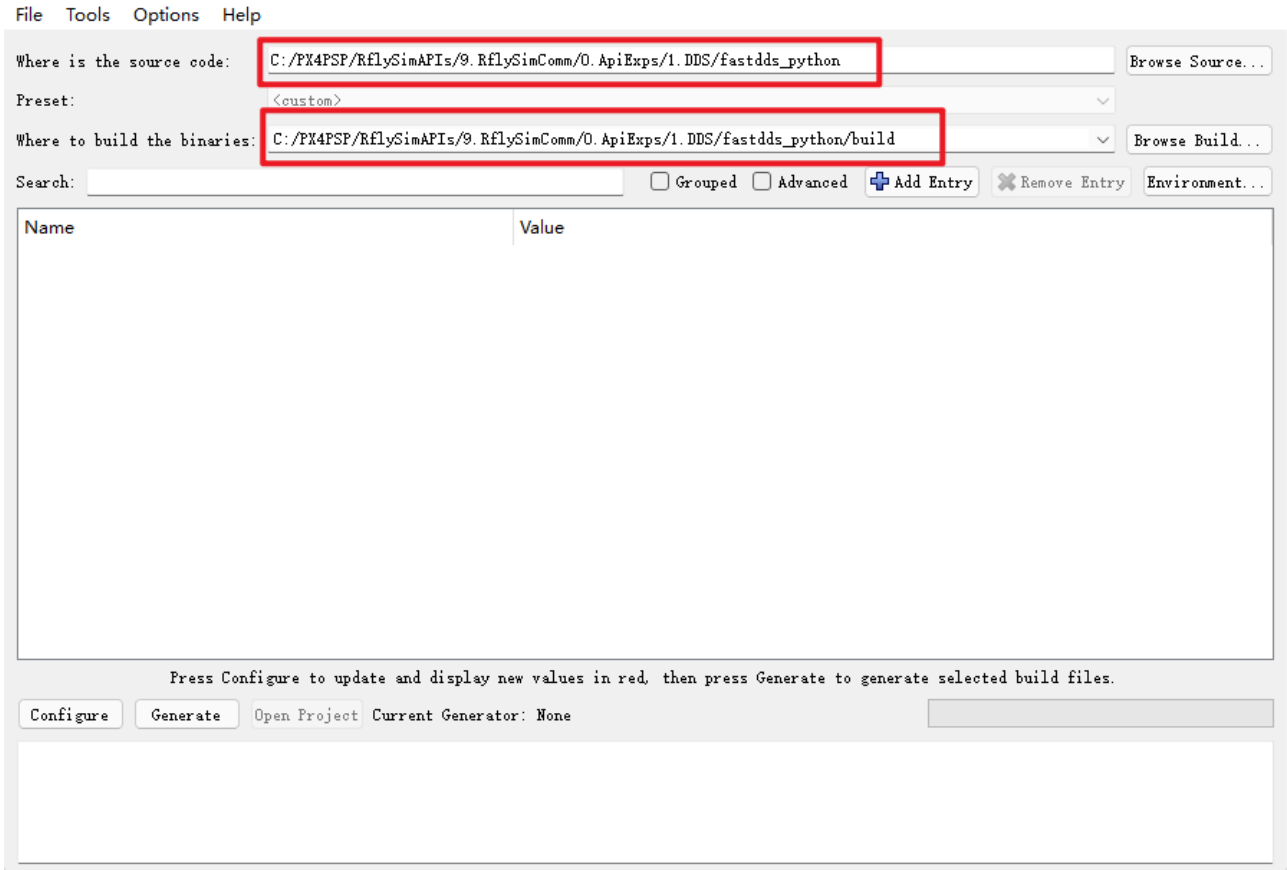
内容包括两部分fastdds库和一个例子，我们首先编译fastdds_python。首先在fastdds_python目录下新建文件夹build。



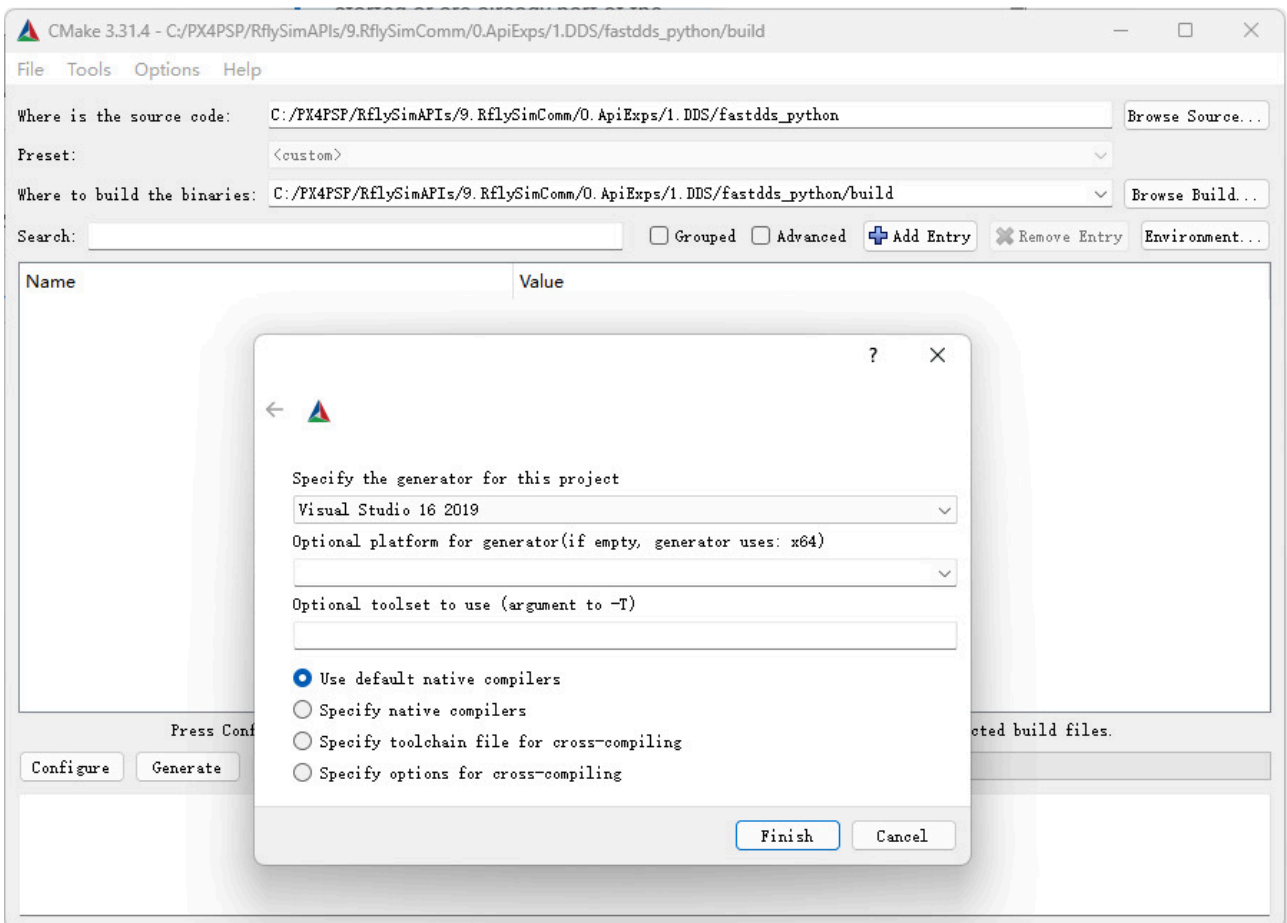
名称	修改日期	类型	大小
build	2023/7/27 14:51	文件夹	
cmake	2023/7/26 17:18	文件夹	
src	2023/7/26 17:18	文件夹	
test	2023/7/26 17:18	文件夹	
CMakeLists.txt	2023/6/29 15:18	文本文档	2 KB
colcon.pkg	2023/6/29 15:18	PKG 文件	1 KB

CSDN @认真的老人

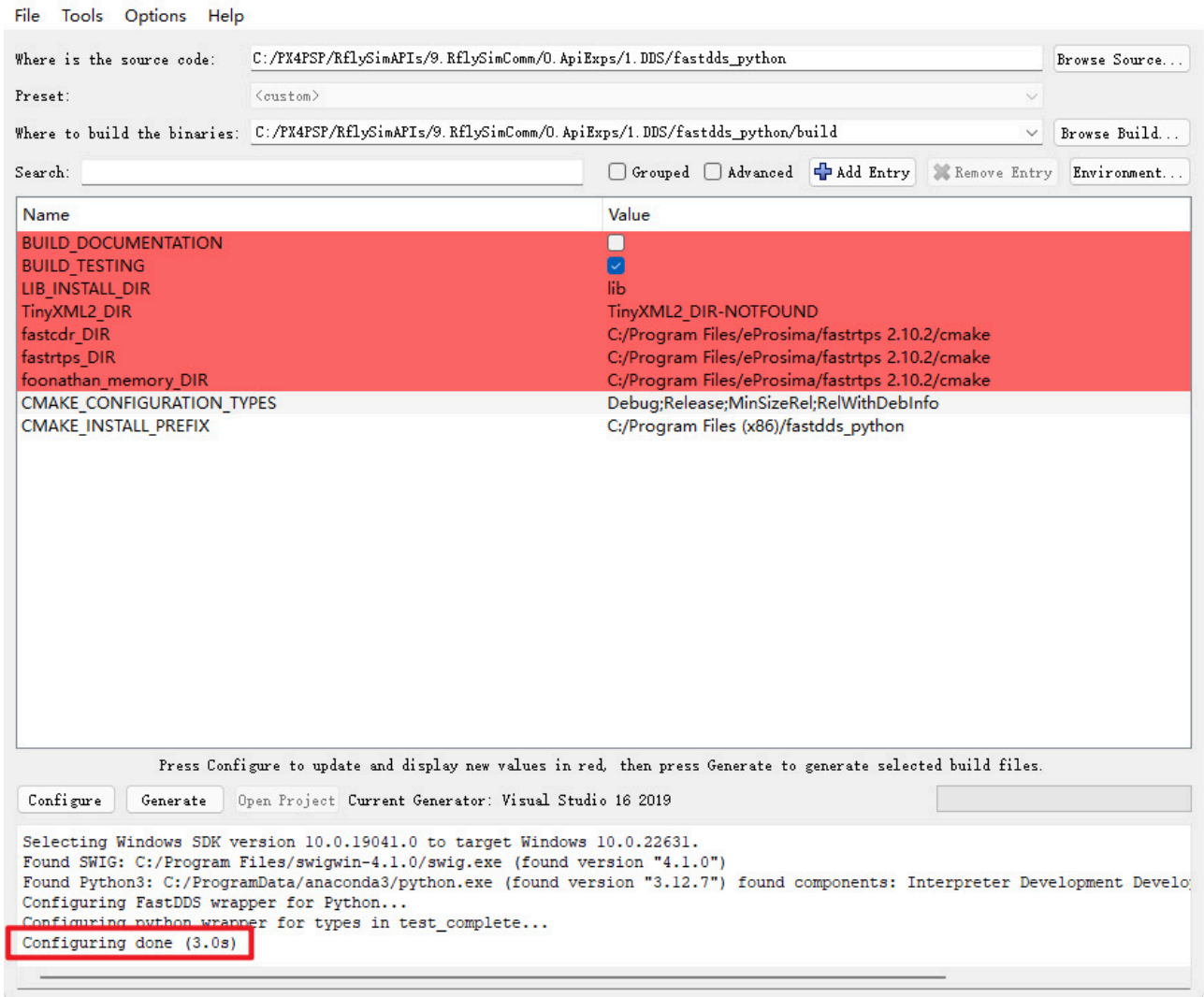
在这里我们使用CMake-GUI工具构建工程，如下图所示，在第一个框中填写fastdds_python目录，在第二个框中填写fastdds_python目录下的build目录：



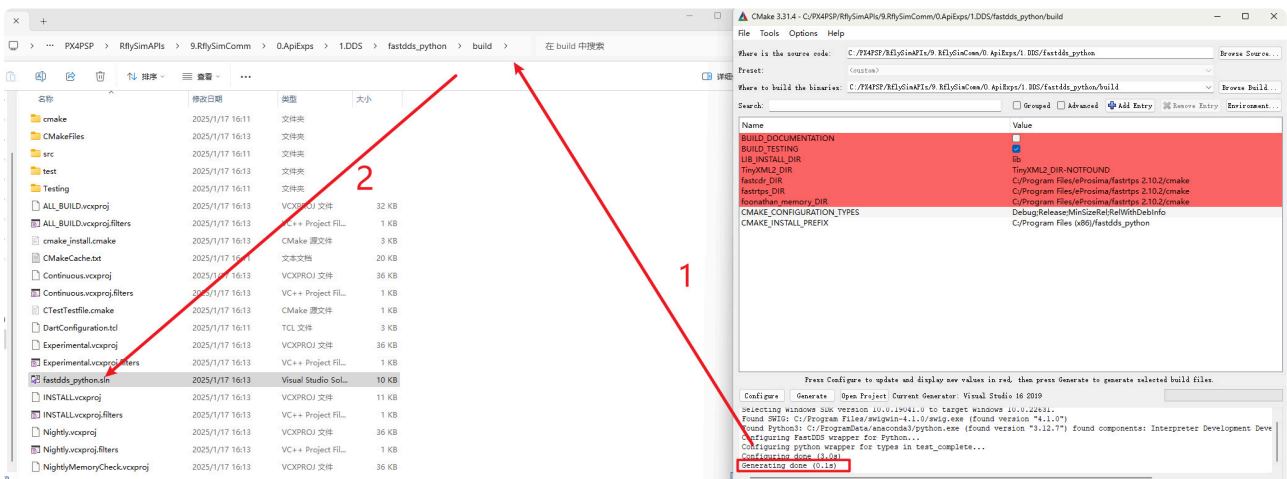
点击Configure后在弹出的对话框中选择上面安装的Visual Studio版本，点击Finish。



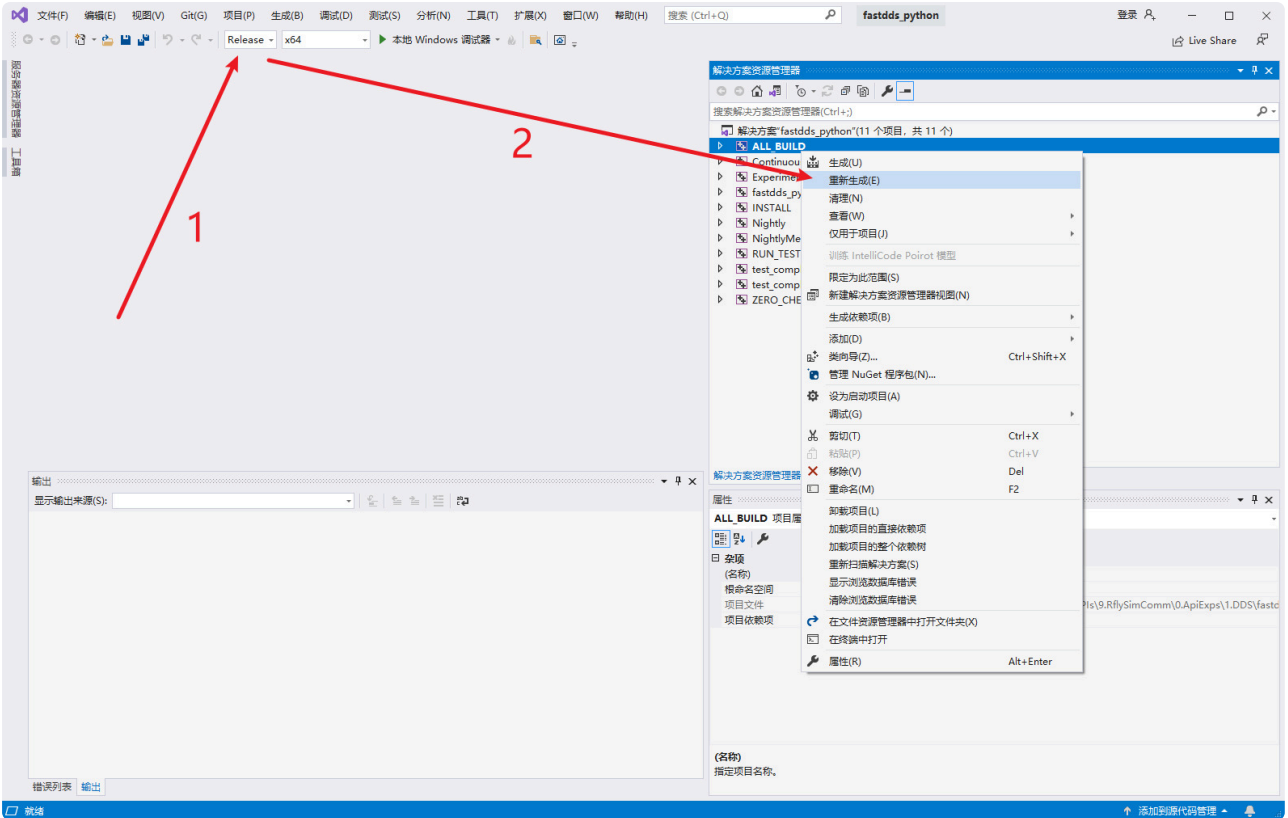
出现Configuring done后，点击Generate:



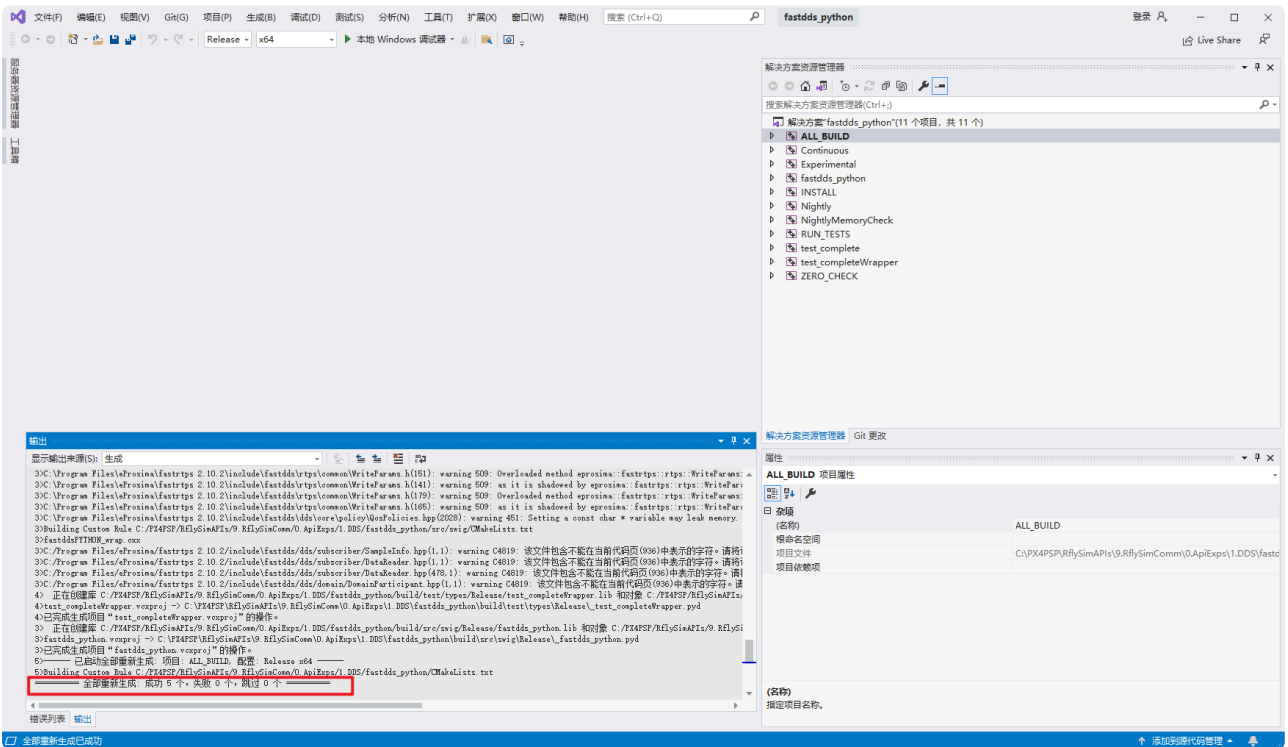
出现Generating done后表示编译通过，此时打开build文件夹，找到fastdds_python.sln文件并双击用Visual Studio打开：



在Visual Studio中将编译模式转成Release，然后在解决方案资源管理器中右键选择重新生成。



出现如下图所示"成功"字样表示编译成功。编译成功后，在build/src/swig下找到 fastdds.py，在build/src/swig/Release下找到_fastdds_python.pyd，这两个文件复制放到一个Demo的空文件夹中用于后面python工程。



名称	修改日期	类型	大小
fastdds_python.pyd	2023/11/10 9:34	PYD 文件	3,943 KB
fastdds_python.exp	2023/11/10 9:34	Exports Library File	1 KB
fastdds_python.lib	2023/11/10 9:34	Object File Library	2 KB

名称	修改日期	类型	大小
CMakeFiles	2023/11/10 9:34	文件夹	
fastdds_python.dir	2023/11/10 9:34	文件夹	
Release	2023/11/10 9:34	文件夹	
cmake_install.cmake	2023/11/9 20:25	CMake 源文件	10 KB
fastdds.py	2023/11/10 9:34	Python 源文件	291 KB
fastdds_python.vcxproj	2023/11/9 20:25	VC++ Project	36 KB
fastdds_python.vcxproj.filters	2023/11/9 20:25	VC++ Project Filter...	2 KB
INSTALL.vcxproj	2023/11/9 20:25	VC++ Project	11 KB
INSTALL.vcxproj.filters	2023/11/9 20:25	VC++ Project Filter...	1 KB
RUN_TESTS.vcxproj	2023/11/9 20:25	VC++ Project	11 KB
RUN_TESTS.vcxproj.filters	2023/11/9 20:25	VC++ Project Filter...	1 KB

4.4 步骤4：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;
};
```

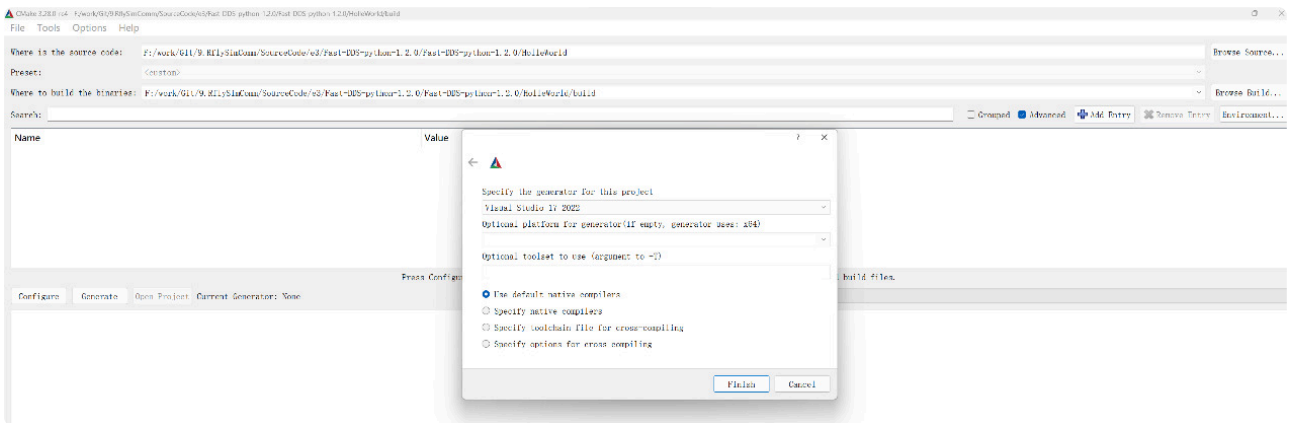
下面是c++头文件的包含。第一个包含HelloWorldPubSubTypes.h文件，其中包含在
新建的文件夹HelloWorld中，打开cmd，并在cmd中使用fastddsgen.bat命令生成
python使用的通信协议文件：

```
fastddsgen.bat -python HelloWorld.idl -ppDisable
```

得到内容如下：

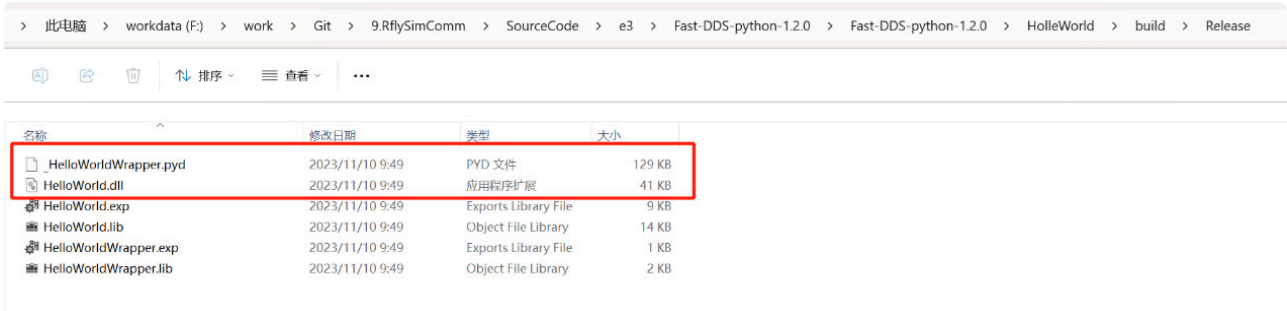
名称	修改日期	类型	大小
build	2023/11/9 20:24	文件夹	
CMakeLists.txt	2023/11/9 18:50	文本文档	4 KB
HelloWorld.cxx	2023/11/9 18:50	QtProject.QtCreato...	5 KB
HelloWorld.h	2023/11/9 18:50	C Header 源文件	7 KB
HelloWorld.i	2023/11/9 18:50	Preprocessed C/C+...	4 KB
HelloWorld.idl	2023/11/9 14:26	Interface Definition ...	1 KB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

在HelloWorld文件夹中重复上述编译操作的过程。

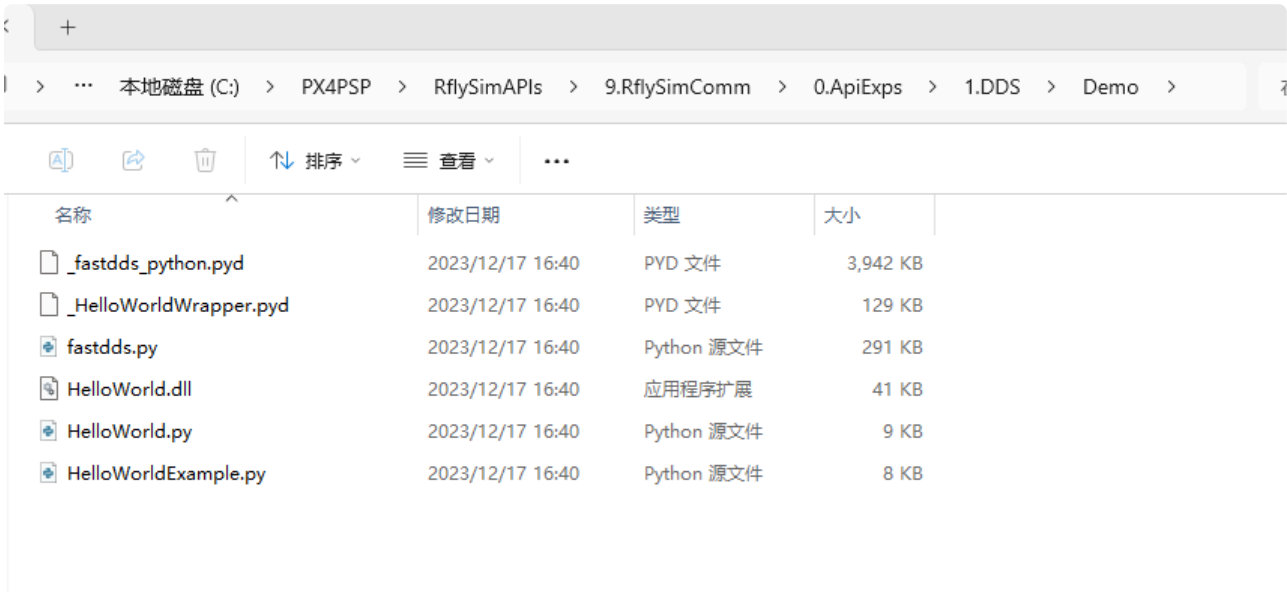


打开VS2022最终生成的文件，后面需要用的文件有_HelloWorldWrapper.pyd、HelloWorld.dll和HelloWorld.py，把这三个文件复制到Demo文件夹中：

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	文本文档	17 KB
HelloWorld.py	2023/11/10 9:49	Python 源文件	9 KB
HelloWorld.sln	2023/11/10 9:30	Visual Studio Soluti...	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



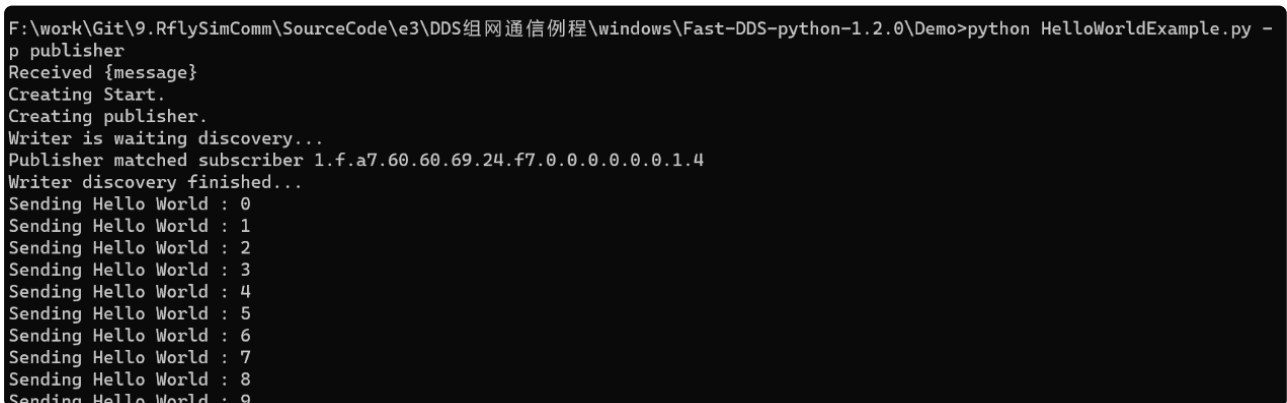
最后，把fastdds_python_example自带例子中的HelloWorldExample.py拷贝至同一个文件夹Demo内。得到如下图内容：



最后在当前路径下打开两个cmd终端，分别执行

```
1 | python HelloWorldExample.py -p publisher
2 | python HelloWorldExample.py -p subscriber
```

得到如图所示效果（可能需要多试几次）：



```
F:\work\Git\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
```

5. 关键知识点

关键知识点1： DDS数据分发服务

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。

关键知识点2： DDS核心原理

DDS的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。

关键知识点3： 自定义协议实现

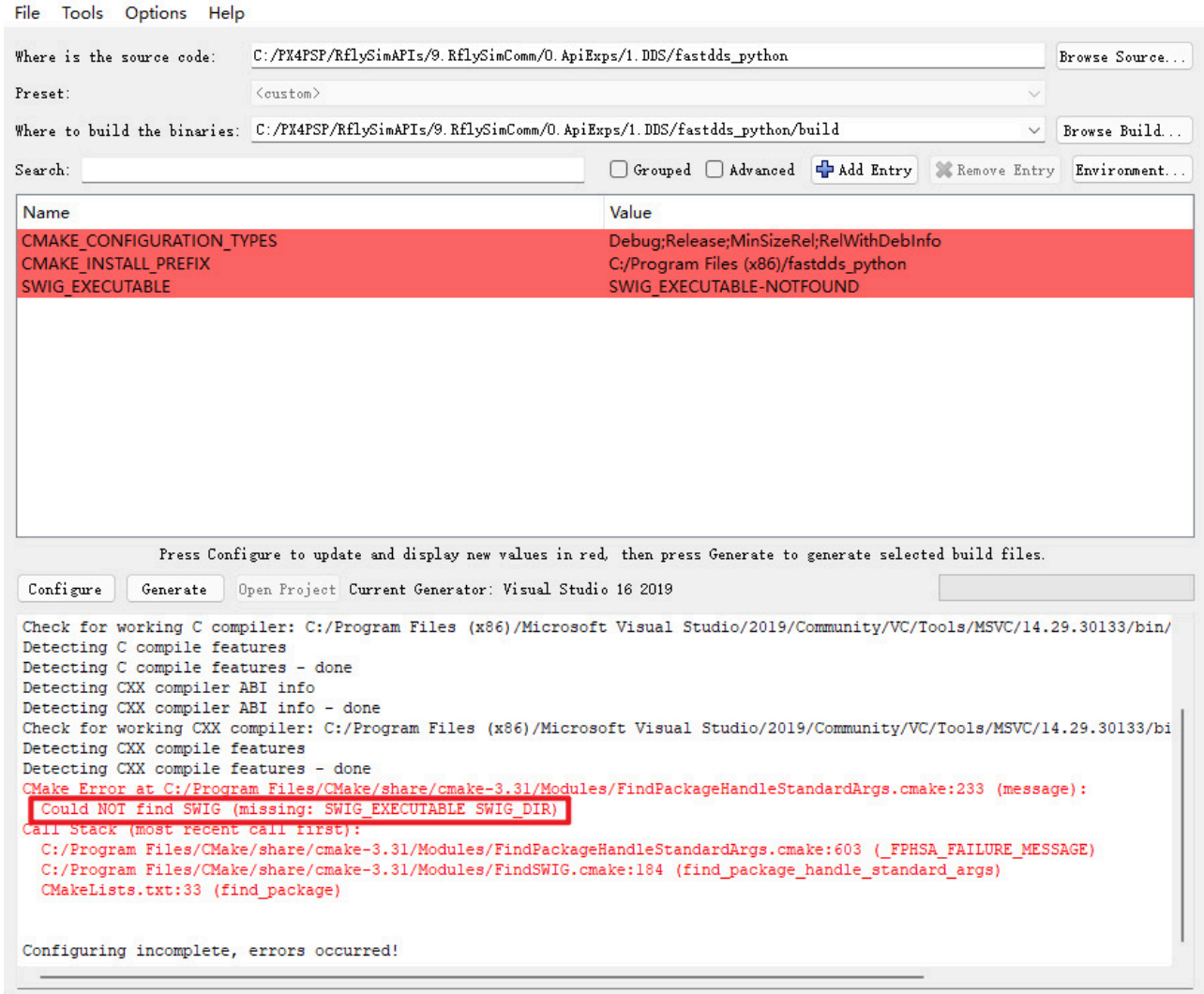
自建协议可能是指基于这些DDS原理，开发出一套符合自身应用需求的具体协议实现，涉及到定制化的数据模型、主题定义、QoS策略及安全保障措施等。

6. 参考资料

1. [RflySim官方文档](#)
2. [Visual Studio: 面向软件开发人员和 Teams 的 IDE 和代码编辑器](#)
3. [Download CMake](#)

7. 常见问题

Q1: CMake报错: Could NOT find SWIG



A1: 请确认SWIG是否安装成功并且环境变量是否正确添加, 参考5.2安装SWIG教程

Q2: 编译过程中出现找不到fastdds模块的错误

A2: 检查环境变量是否正确设置, 特别是FASTRTPSHOME和PATH变量是否包含了Fast DDS的安装路径。

Q3: 运行HelloWorldExample时无法建立通信连接

A3: 确保在同一网络下运行publisher和subscriber, 且防火墙未阻止相应的端口通信。

1. <https://rflysim.com/> ↩

2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩