JAKA 节卡

JK-RDQK-UM-029

JAKA QUICK START FOR SECONDARY DEVELOPMENT

上海节卡机器人科技有限公司　Shanghai JAKA Robotics Ltd
电话 Tel ：+400 006 2665 | 网站 Web:www.jaka.com
上海：上海市闵行区剑川路610号33-35栋 | Building 33-35,No.610 Jianchuan Rd, Shanghai
常州：江苏省常州市武进国家高新区武宜南路 377 号 10 号楼 | Building 10, No.377 South Wuyi Rd, Changzhou, Jiangsu

Version Record

Version No.	Version Date	Description
V1.0	2020.07.12	Create document

Content

1. Introduction

JAKA secondary development is based on the network communication protocol TCP/IP to communicate with the robot, with an interface for robotic manipulation and supporting four programming languages, python, C/C++, and C#. The secondary development material package mainly provides simulation models of different types of JAKA robots on different simulation platforms, and it also provides JAKA SDK documents and examples of different programming languages.

2. Recommended Reading Material

JAKA Secondary Development FAQ (Know about some FAQ)
Quick Start-Electrical Cabinet V2.1 (Use JAKA APP operation robot through this document)
JAKA_SDK Material Package\Description Document (SDK documents of different programming language and dynamic library usage document)

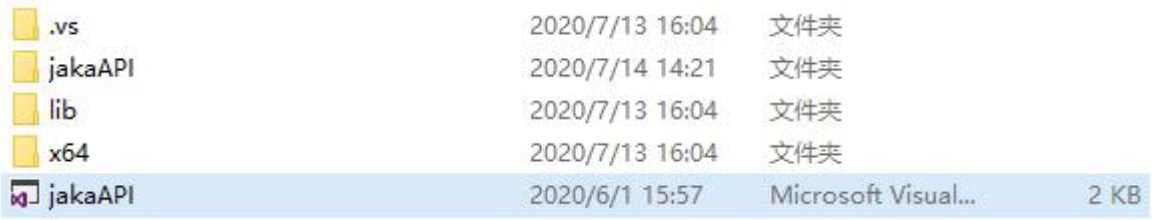
3. C++ SDK Quick Start

Development environment:**Windows x64, Vs2013**

Document: **JAKA secondary development material package\JAKA_SDK material package\Description document\JAKA secondary development document c++.pdf**

Example: **JAKA secondary development material package\JAKA_SDK material package\example\windows dynamic library examples project\c++\projectjakaAPI.sln**

As shown in the following figure:



After using vs to open project examples, you need to pay attention to modify the IP to the robot IP used by yourself when using login_in() interface in main() function, and then running the program, but pay attention that there is no interference to avoid collisions:

```
int main()
{
    //Instantiate API demo
    JAKAZuRobot demo;
    //Log in controller, replace 192.168.2.229 with robot IP that you are currently using
    demo.login_in("192.168.2.229");
    //Power on the robot
    demo.power_on();
    //Enable the robot
    demo.enable_robot();
```

Refer to following documents to know specific usage:

JAKA JAKA secondary development material package\JAKA_SDK material package\Description document\JAKA secondary development document c++.pdf

JAKA secondary development material package\JAKA_SDK material package\Description document\Dynamic library usage instructions.pdf

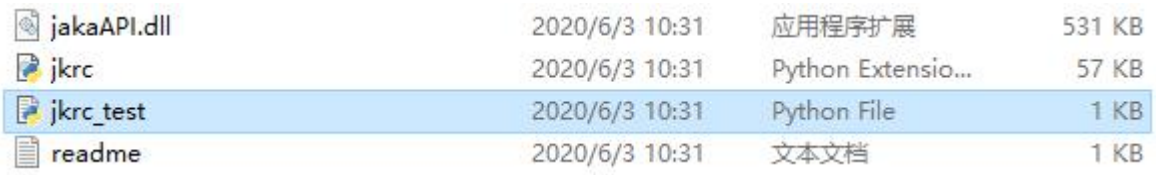
4. Python Quick Start

Development environment:Windows x64, Python3.7.0

Document: JAKA secondary development material package\JAKA_SDK material package\Description document\JAKA secondary development document Python.pdf

Example:JAKA secondary development material package\JAKA_SDK material package\ python\jkrc_windows_x64_py3.7 usage instructions\ jkrc_test.py

As shown in the following figure:



Notes:

1. Add the path of jkrc to ensure that the Python interpreter can find the jrkc module.
2. Replace the robot ip with the robot ip you are currently using.

```
import sys
import time
sys.path.insert(0, 'F:\\xx\\xx\\xx\\xx')  #Ensure that this path is searched firstly, and the user can replace it as needed
                                          #Otherwise Python interpreter may not find jkrc module
import jkrc
#COORD_BASE = 0      # Base coordinate system
#COORD_JOINT = 1     # Joint space
#COORD_TOOL = 2      # Tool coordinate system
robot = jkrc.RC("192.168.2.194")#Replace the IP with robot IP that you are using currently
robot.login()  # Log in
ret = robot.get_joint_position()
if ret[0] == 0:
    print("the joint position is :",ret[1])
else:
    print("some things happend,the errcode is: ",ret[0])
robot.logout()  # Log out
```

Refer to following documents to know specific usage:

JAKA secondary development material package\JAKA_SDK material package\Description document\JAKA secondary development document Python.pdf

5. Dynamic Library Usage Instructions

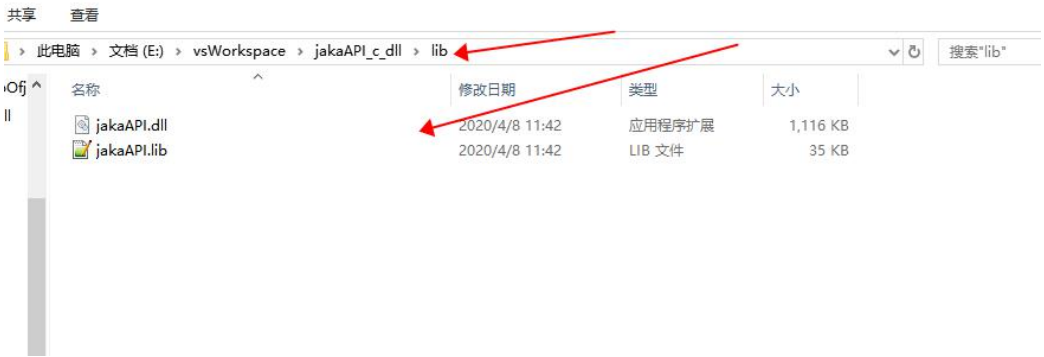
1. **This instruction describes how to use jakaAPI.dll in visual studio.**
2. **Version requirements**

The project uses visual studio 2013 in windows, the compiler is MSVC-120, and the minimum supported compiler version is MSVC-120.

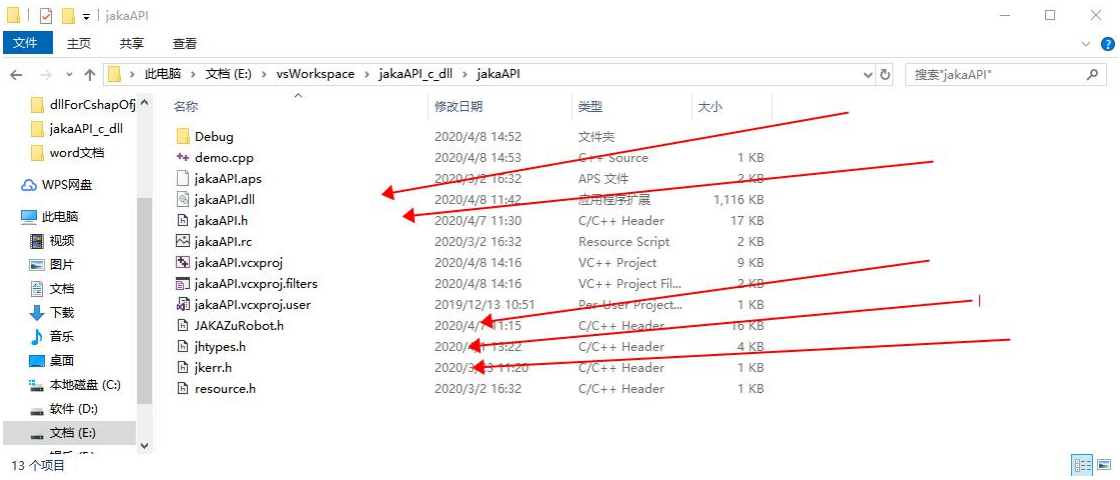
The minimum gcc compiler version is 6.3.0 in linux dynamic library.

3. **Add dynamic library（C/C++） in windows**

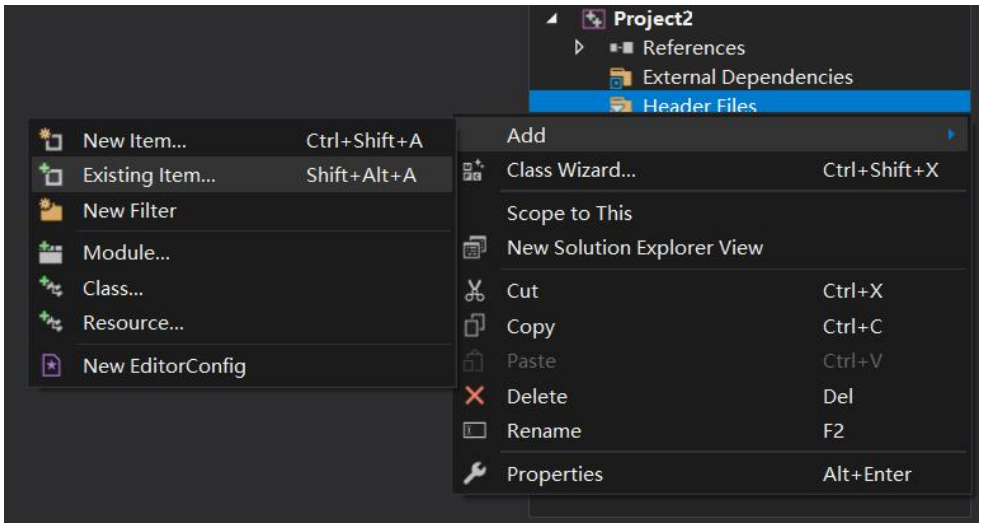
The dynamic library required by the project should be placed in a folder, take lib folder as an example(as shown in the following figure), choosing different dynamic libraries according to different platforms, this project takes debug, x86 as examples.



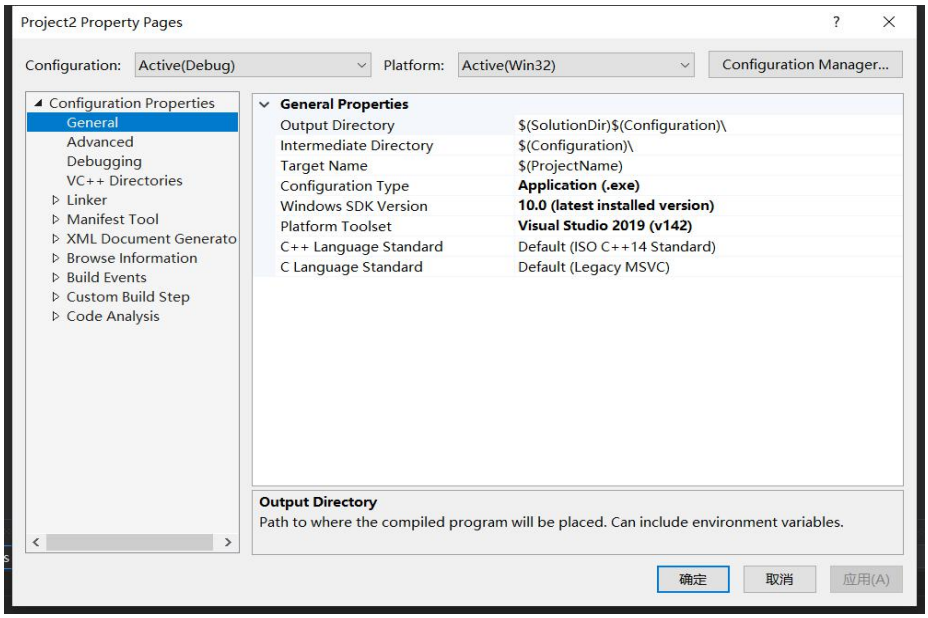
Put the required header files and dll files in a folder. In this project, put the header files in the project folder.



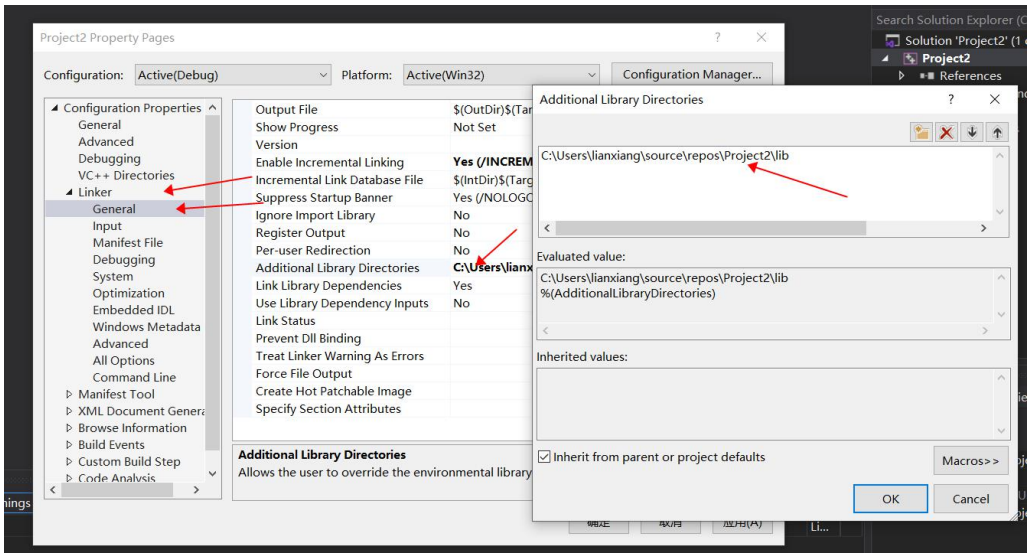
Right-click the "Header files" and select "Add" --- "Existing Item" to add the required header file.



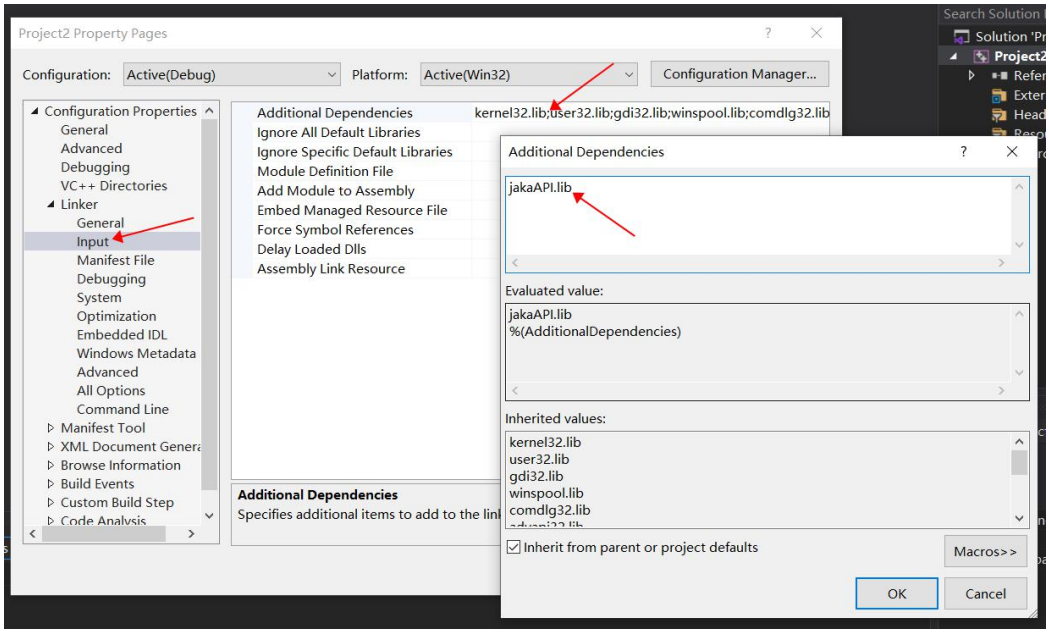
Right-click the solution, select properties, and open the properties configuration tab.



Select "Linker"-"General"-"Additional Library Directories" to add the path of dynamic library.



Select "Linker"-"Input"-"Additional Dependencies" to add the name of dynamic library.



4. Add dynamic library（C/C++） in Linux

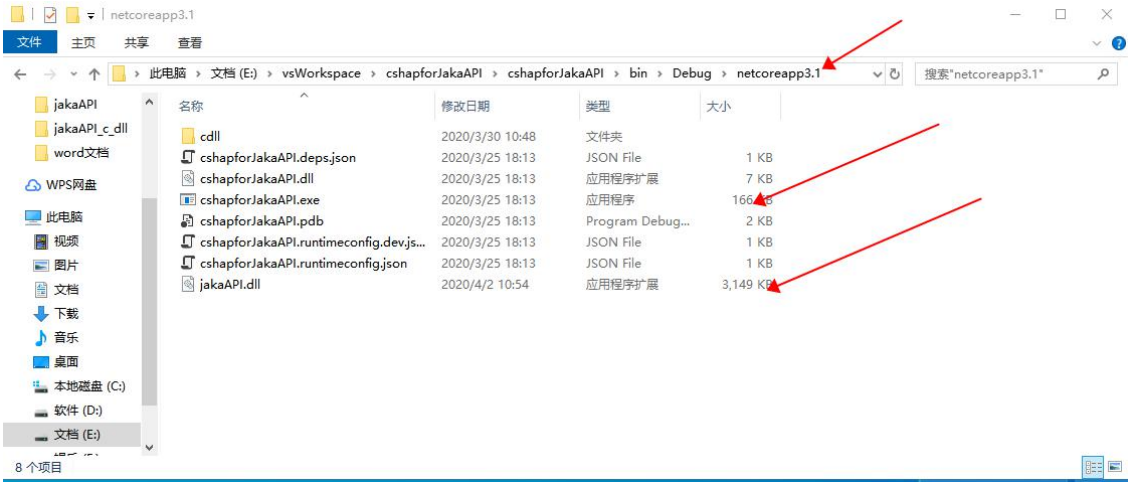
(1) Link the path of dynamic library to the project

(2) Link the path of document to the project

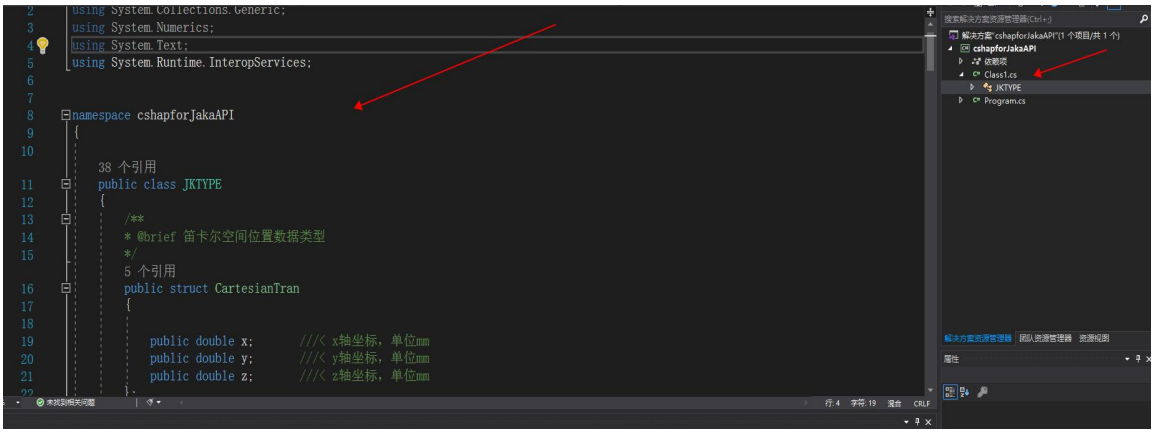
Two example projects are provided here using cmake compiler (32-bit) and QT compiler (34-bit) respectively. Please refer to <Linux dynamic library example project>.

5. Add dynamic library（C#）

Put the dll file in the same path as the exe file.



Create a new data structure file to define the necessary data structure. These structures are already defined in "jaka secondary development\jaka secondary development dynamic library and documentation\dynamic library\windows\c# dynamic library and usage instructions\function structure.txt" , just copy and paste it directly.



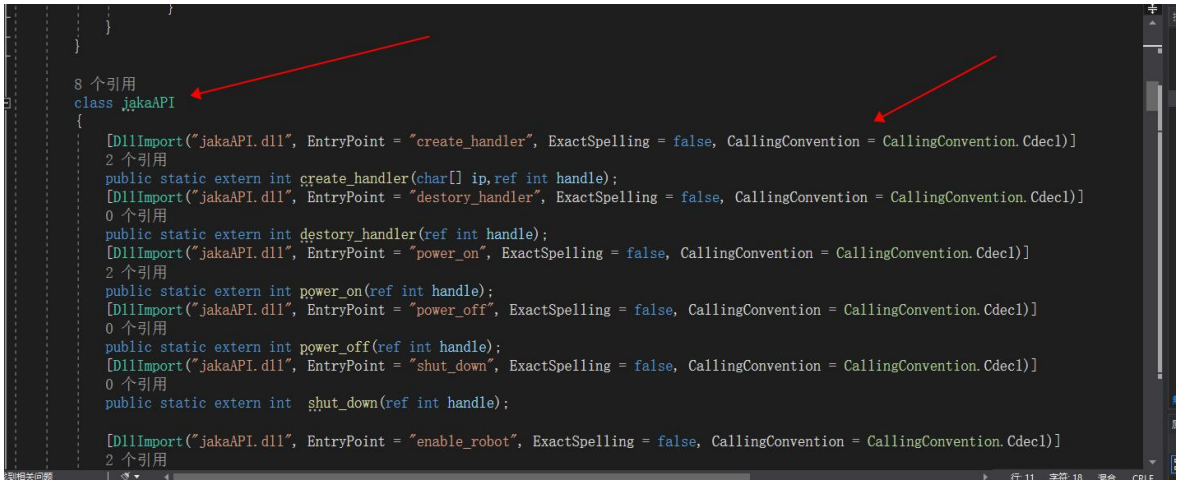
```
using System.Collections.Generic;
using System.Numerics;
using System.Text;
using System.Runtime.InteropServices;


namespace cshapforJakaAPI
{

    38 个引用
    public class JKTYPE
    {
        /**
        * @brief 笛卡尔空间位置数据类型
        */
        5 个引用
        public struct CartesianTran
        {

            public double x;      ///< x轴坐标，单位mm
            public double y;      ///< y轴坐标，单位mm
            public double z;      ///< z轴坐标，单位mm
```

Declare the function call in jakaAPI.dll in the main program. These declarations are already defined in "jaka secondary development\jaka secondary development dynamic library and documentation\dynamic library\windows\c# dynamic library and usage instructions\method declaration.txt", just copy and paste it directly.



```
    }
}

8 个引用
class jakaAPI
{
    [DllImport("jakaAPI.dll", EntryPoint = "create_handler", ExactSpelling = false, CallingConvention = CallingConvention.Cdecl)]
    2 个引用
    public static extern int create_handler(char[] ip,ref int handle);
    [DllImport("jakaAPI.dll", EntryPoint = "destory_handler", ExactSpelling = false, CallingConvention = CallingConvention.Cdecl)]
    0 个引用
    public static extern int destory_handler(ref int handle);
    [DllImport("jakaAPI.dll", EntryPoint = "power_on", ExactSpelling = false, CallingConvention = CallingConvention.Cdecl)]
    2 个引用
    public static extern int power_on(ref int handle);
    [DllImport("jakaAPI.dll", EntryPoint = "power_off", ExactSpelling = false, CallingConvention = CallingConvention.Cdecl)]
    0 个引用
    public static extern int power_off(ref int handle);
    [DllImport("jakaAPI.dll", EntryPoint = "shut_down", ExactSpelling = false, CallingConvention = CallingConvention.Cdecl)]
    0 个引用
    public static extern int  shut_down(ref int handle);

    [DllImport("jakaAPI.dll", EntryPoint = "enable_robot", ExactSpelling = false, CallingConvention = CallingConvention.Cdecl)]
    2 个引用
```

6.Feedback and Errata

For any inaccurate descriptions or errors in the document, we would like to invite the readers to correct and criticize. In case of any questions during your reading process or any comments you want to make, please send an email to support@jaka.com, and our colleagues will try to reply one by one.

Property Description