



PQ Kit 部署说明文档

文档版本: V3.3
修订时间: 2023.07.26



产权说明

注意：此处包含的所有信息均为华航唯实机器人科技股份有限公司的财产。未经华航唯实机器人科技股份有限公司的事先书面许可，本出版物的任何部分（无论是硬拷贝还是电子形式）均不得复制、存储在检索系统中，或以任何形式或通过任何方式如电子、机械、影印、录音、或以其他方式传输。请注意，即使未与包含最终用户许可协议的软件一起分发，本指南中的内容仍受版权法保护。PQ 和 PQArt 的标志是华航唯实机器人科技股份有限公司的注册商标。所有其他商标均为其各自所有者的财产。

本出版物和此处的信息按原样提供，仅供参考，如有更改，恕不另行通知，且不应被视为华航唯实公司的承诺。华航唯实公司明确声明对可能出现在本指南所载信息中的任何错误或不准确之处不承担任何责任或义务，对本文件不作任何（明示、暗示或法定）保证，并明确表示不承担任何关于适销性、适用于特定目的和不侵犯第三方权利的保证。



更新记录

版本编号	版本日期	PQ Kit 支持版本	更新说明
V3.3	2023-07-26	9.1.0.6105 以及上	原 2.13 拆成 2.13、2.14，新增 2.2



版本说明

PQKit 分为开发版、部署版两种版本。开发版为开发者所用版本，部署版为用户打包部署自己开发完的上层应用所用版本。

开发版与部署版的区别如下：

- 开发版 SDK 安装包中除了 PQKit 组件文件以外还包含了各种编程语言下集成 PQKit 的 Sample 源码及可执行文件、PQKit 部署文档、PQApi 手册等。部署版仅提供客户安装部署可用的 PQKit 组件文件安装包。
- 开发版 PQKit 仅供开发过程使用，可以使用第三方开发工具进行开发、调试等开发工作。部署版仅供程序运行使用。
- 开发版除了提供离线编程应用技术方面技术支持外，还提供开发技术咨询服 务，部署版不提供技术支持。



1. PQ Kit 安装部署

1.1 安装

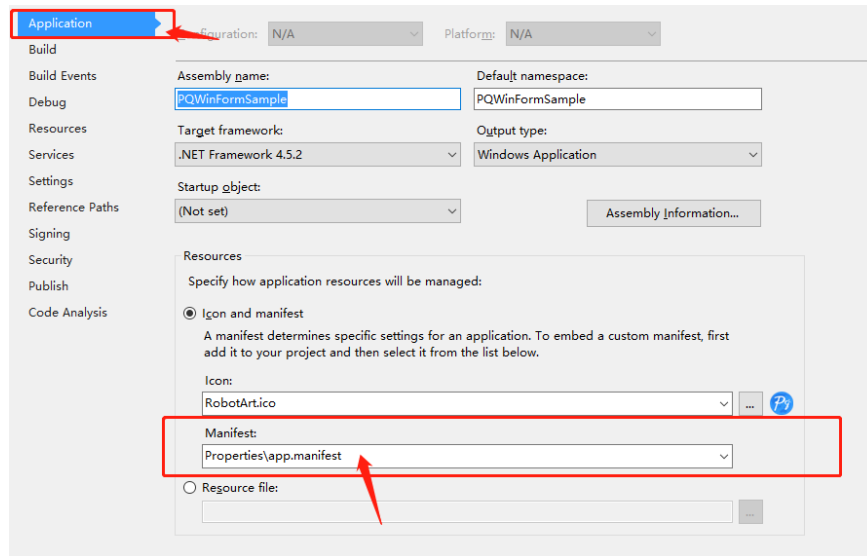
- ✧ **PQ Kit 为内嵌式开发组件，需要被上层应用集成，无法单独运行。**
- ✧ 上层应用 exe 所在目录必须为 "bin"
- ✧ PQ Kit 除了双击安装包直接安装外，支持以命令行指定路径、静默参数安装。例如，上层应用 exe 所在目录 bin 的上层目录为 D:\test 且需要静默安装，则命令行需传参 /p="D:\test" /S。



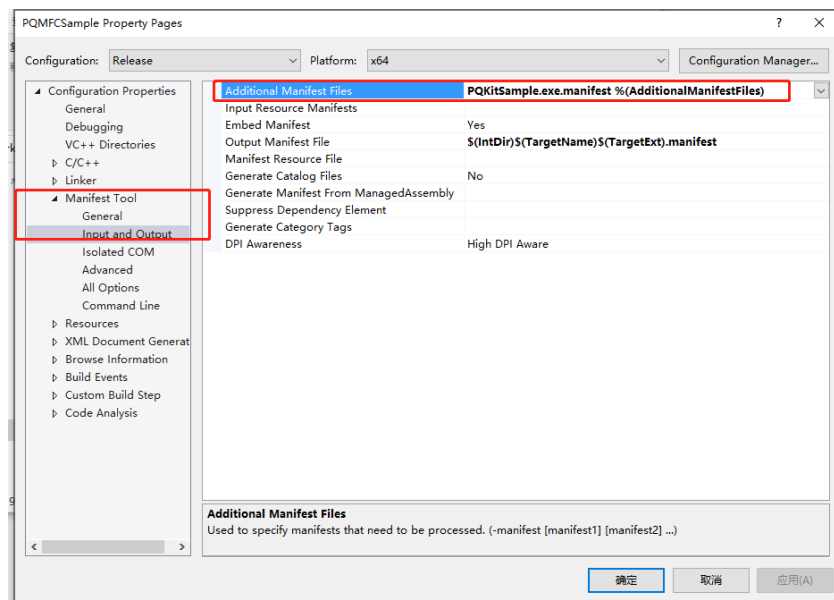
图 1 命令行安装 SDK

1.2 部署

- ✧ 上层应用编译生成结果需置于 bin 目录下。
 - ✧ 上层应用开发工具最低要求为 **Visual Studio 2015**。
 - ✧ 自行创建 **C#工程**需要在工程属性 Application-Icon and manifest 处指定 Manifest 文件，需要将系统自动生成的 Properties\app.manifest 文件替换成 Sample\include\C#下的 app.manifest。
- a. Sample 工程已处理好属性设置，直接编译即可。
- b. 新建 C# 工程需在 Properties-Application-Security 处勾选 Enable ClickOnce security settings 方可指定 Manifest 文件，指定 Manifest 后需取消勾选 Enable ClickOnce security settings。



- ✧ 自行创建 C++工程需要在工程属性 Manifest Tool-Input and Output 处指定 Additional Manifest Files 文件，PQKitSample.exe.manifest 文件放在 Sample\include\C++下。(Sample 工程已处理好属性设置，直接编译即可。)





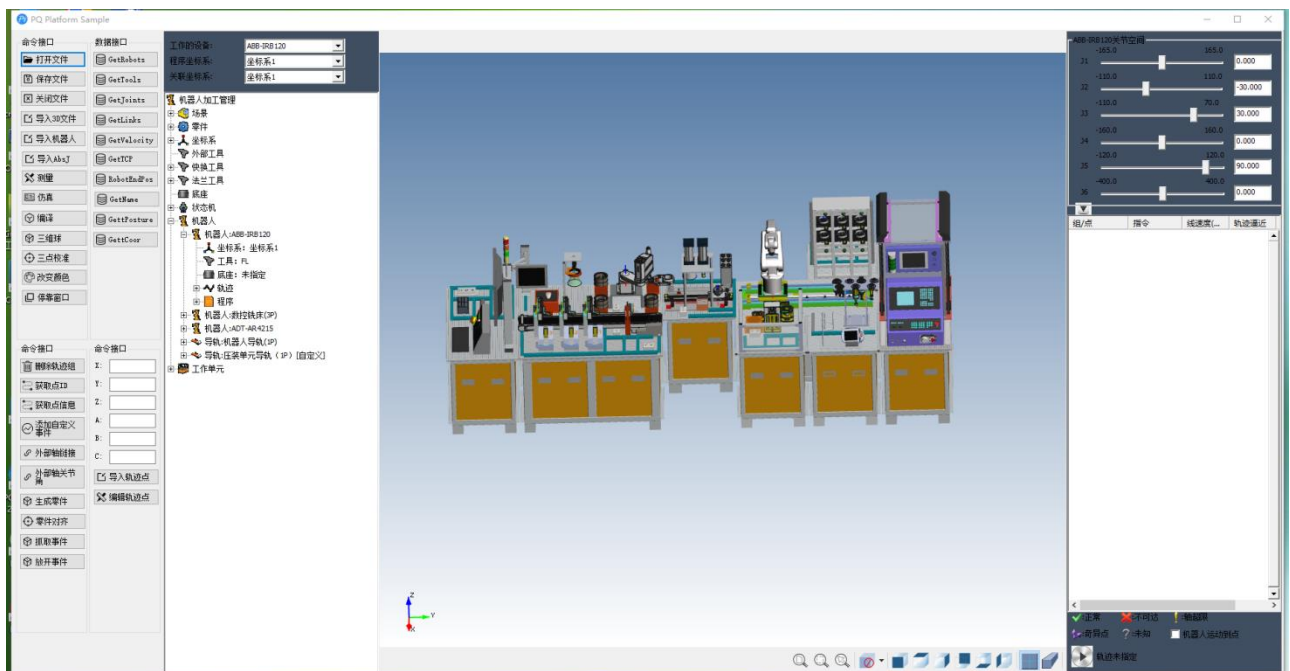
1.3 Sample

PQ Kit 会在安装目录下打入 Sample 目录,其中包含 C#示例 PQWinFormSample, C++示例 PQMFCSample 以及示例工程需要的相关文件(include)。

安装会在安装目录 bin 下打包已编译好的 sample 可执行文件并在桌面创建相应的快捷方式,直接打开即可。

名称	修改日期
PQMFCSample.exe	2022/10/28 15:4
PQWinFormSample.exe	2022/10/28 16:3
PQWinFormSample.exe.manifest	2022/10/28 17:0
TKPQK100.dll	2022/10/29 1:21
UnrealEngine.dll	2022/10/29 1:22
UnrealLauncher.dll	2022/10/29 1:23
UP3DSceneData.dll	2022/10/29 1:23
UnrealUI.dll	2022/10/29 1:32
UPC.dll	2022/10/29 1:32
UP3DSceneUI.dll	2022/10/29 1:32
UP3DSceneGui.dll	2022/10/29 1:32
UPApiRawData.dll	2022/10/29 1:33
UPApiRawCam.dll	2022/10/29 1:33
UPApiRawUI.dll	2022/10/29 1:37
UPApiBase.dll	2022/10/29 1:35

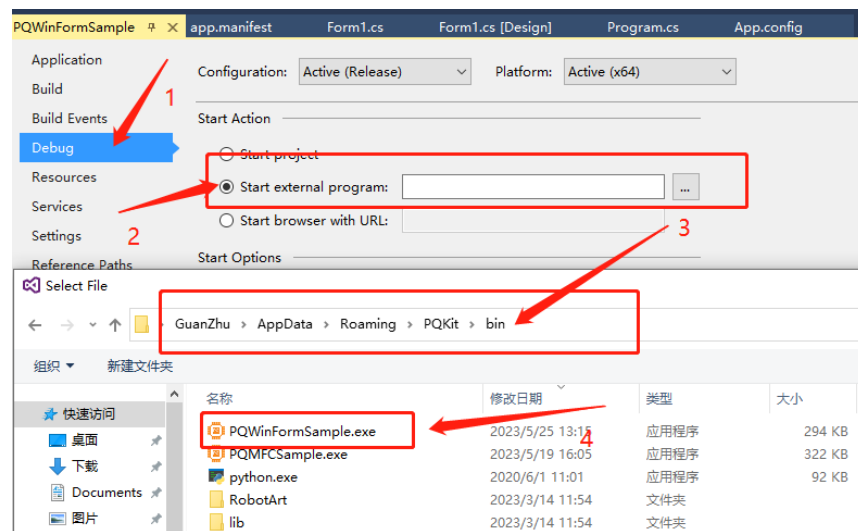
其中 C#示例效果如下图所示





1.3.1 Sample 调试

PQKit 调试需要指定外部程序，即拷贝至 PQKit 安装目录 bin 下的用户上层程序。以 C#工程为例，调试设置如下图所示。



2. PQ Kit 初始化

2.1 初始化 API

2.1.1 初始化

```
HRESULT pq_InitPlatformComponent(IPQPlatformComponentCallback* pCallback, int hParentHwnd, BSTR bsName, BSTR bsPWD)
```

Parameters:

pCallback	回调接口指针，用于接收初始化状态
hParentHwnd	用于显示组件的窗口句柄
bsName	登录 kit 的用户名
bsPWD	登录 kit 的用户密码

Return:

返回值从 IPQPlatformComponentCallback 接口派生的对象方法 Fire_Initialize_Result 中获取，各个返回值状态如下：

1	初始化成功
-1	一般性错误
-2	初始化超时



-3、-4、-6	加载组件动态库错误
-5	用户校验失败
-7	组件重复初始化
-9、-10、-11、-12	内部数据错误

2.1.2 状态回调

初始化回调

void Fire_Initialize_Result(int lResult)	
Parameters:	
lResult	Kit 初始化状态返回值
返回值状态如下:	
1	初始化成功
-1	一般性错误
-2	初始化超时
-3、-4、-6	加载组件动态库错误
-5	用户校验失败
-7	组件重复初始化
-9、-10、-11、-12	内部数据错误

用户登录状态回调

void Fire_Login_Result (int lResult)	
Parameters:	
lResult	Kit 登录状态返回值
返回值状态如下:	
1	登录成功
-1	密码错误
-2	用户不存在
-4	用户已过期
-6	网络连接失败
-7	用户未激活
-14	账号不匹配

2.1.3 获取三维显示区域

获取 PQ Kit 三维显示主区域。



HRESULT pq_GetPlatformView(int* hView)

Parameters:

hView	返回 Kit 三维显示区域句柄
--------------	-----------------

2.1.4 默认停靠窗口

PQ Kit 中机器人加工管理树、调试窗口、输出窗口、机器人控制窗口属于默认停靠窗口。这 4 个窗体有两种显示方式，其一是内嵌模式，即与 PQArt 软件行为一致，停靠在主三维区域；其二是浮动模式，即用户可获取窗口句柄自行显示在自己的应用窗体中。

PQ Kit 默认行为为浮动模式，同时提供了接口设置改参数。用户可通过接口 PQAPISetOption 设置浮动还是内嵌。即 PQKitOption 结构体中 nEmbedded 置 0 则浮动，置 1 则内嵌。

a. 机器人加工管理树

HRESULT pq_GetModelTreeView (int* hView)

Parameters:

hView	返回机器人加工管理树句柄
--------------	--------------

b. 调试窗口

HRESULT pq_GetDebugTreeView (int* hView)

Parameters:

hView	返回调试窗口句柄
--------------	----------

c. 输出窗口

HRESULT pq_GetOutPutView (int* hView)

Parameters:

hView	返回输出窗口句柄
--------------	----------

d. 机器人控制窗口

HRESULT pq_GetRobotControlView (int* hView)

Parameters:



hView

返回机器人控制窗口句柄

2.2PQ Kit Option

用户可通过 PQAPISetOption 来设置一些选项参数。

HRESULT PQAPISetOption(PQKitOption *i_stuOption)

Parameters:

i_stuOption

可设置的选项参数结构体数据

结构体定义如下

```
typedef struct struct_PQKitOption
{
    int nEmbedded;
}PQKitOption;
```

目前支持的选项参数如下:

Parameters:

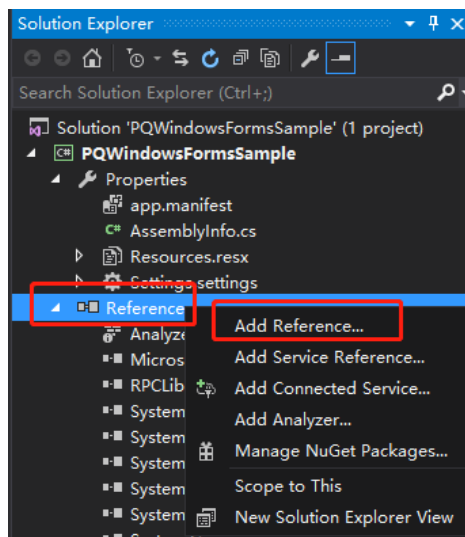
nEmbedded

决定默认停靠窗口是浮动还是内嵌，置 0 则浮动，置 1 则内嵌

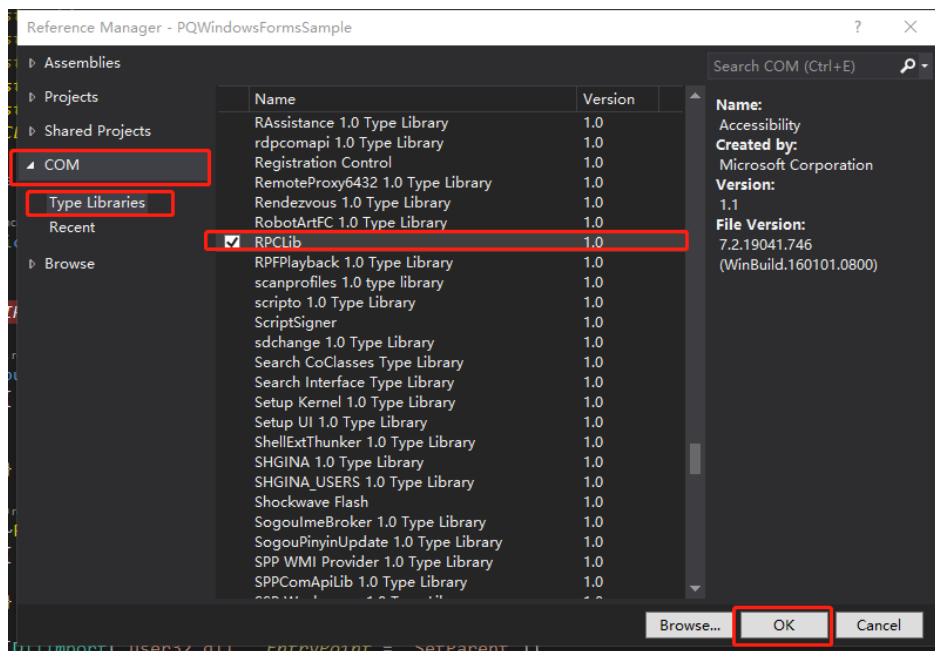
2.3 C#范例

2.2.1 新建 C# WinForm 工程。

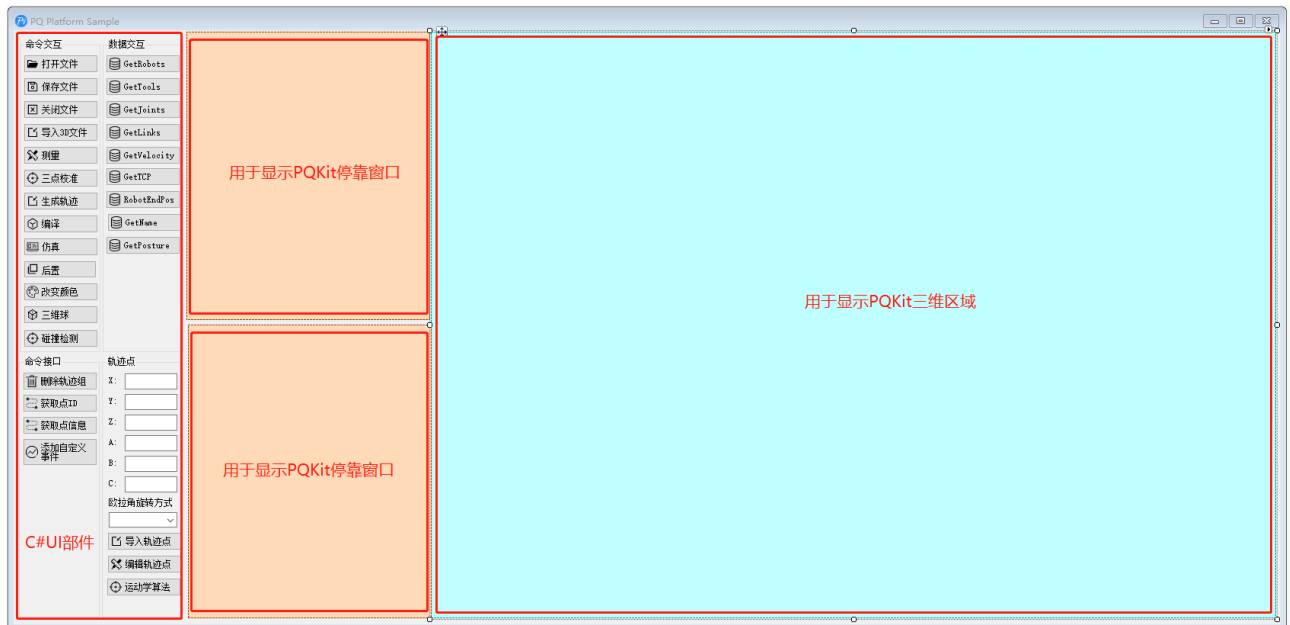
2.2.2 在工程解决方案 References, 右键 Add Reference...



依次选择 COM-Type Libraries-RPCLib, 单击 OK 添加 COM 组件。



2.2.3 在窗体设计中, 拖入一个 Panel 或者其他 Container 作为组件的显示容器。



2.2.4 在窗体 Load 函数中新建 IPQPlatformComponent 接口对象、回调接口对象 PQPlatformComponentCallBack，启动线程 InitializeKit。

```
public delegate void OnKitInitialized();  
//  
IPQPlatformComponent m_ptrKit = null;  
//  
PQPlatformComponentCallBack m_CallBack = null;  
Form1()  
{  
    System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = false;  
    this.WindowState = FormWindowState.Minimized;  
    InitializeComponent();  
}  
private void Form1_Load(object sender, EventArgs e)  
{  
    //创建内核对象  
    m_ptrKit = new PQPlatformComponent();  
    //创建callback对象  
    m_CallBack = new PQPlatformComponentCallBack();  
    m_CallBack.SetMainWindow(this);  
    //内核启动需要在子线程中  
    new Thread(InitializeKit).Start();  
}
```

在 InitializeKit 线程函数中调用 pq_InitPlatformComponent 初始化组件，需要输入用户名、密码(用户名、密码需要申请开通)。



```
private void InitializeKit()
{
    //此处为内核组件的初始化，需依次传入内核组件回调接口，用于显示内核组件的句柄，用户名、密码
    int nHWND = (int)panel1.Handle;
    string strPath = this.GetType().Assembly.Location;
    string strName = "";
    string strPWD = "";
    m_ptrKit.pq_InitPlatformComponent(m_Callback, nHWND, strName, strPWD);
}
```

在 PQPlatformComponentCallback 类 Fire_Initialize_Result 方法中获取初始化状态。如果初始化成功则将 WinForm 中的显示容器设为组件显示区域的 Parent 并移动位置，如果初始化失败，查看失败错误码具体分析即可。

```
public void Fire_Initialize_Result(int lResult)
{
    if (lResult > 0)
    {
        mainWindow.Invoke(new OnKitInitialized(mainWindow.ShowPlarformWindow));
    }
}
//C# Form类中显示组件方法
public void ShowPlarformWindow()
{
    //此处将内核组件显示在C#窗体的句柄中
    //由回调接口Fire_Initialize_Result函数中按初始化结果判定调用与否
    int nHWND = 0;
    m_ptrKit.pq_GetPlatformView(out nHWND);
    IntPtr child = (IntPtr)nHWND;
    SetParent(child, panel1.Handle);

    MoveWindow(child, 0, 0, panel1.Width, panel1.Height, true);

    ShowDockWindow();

    this.WindowState = FormWindowState.Normal;
}
```

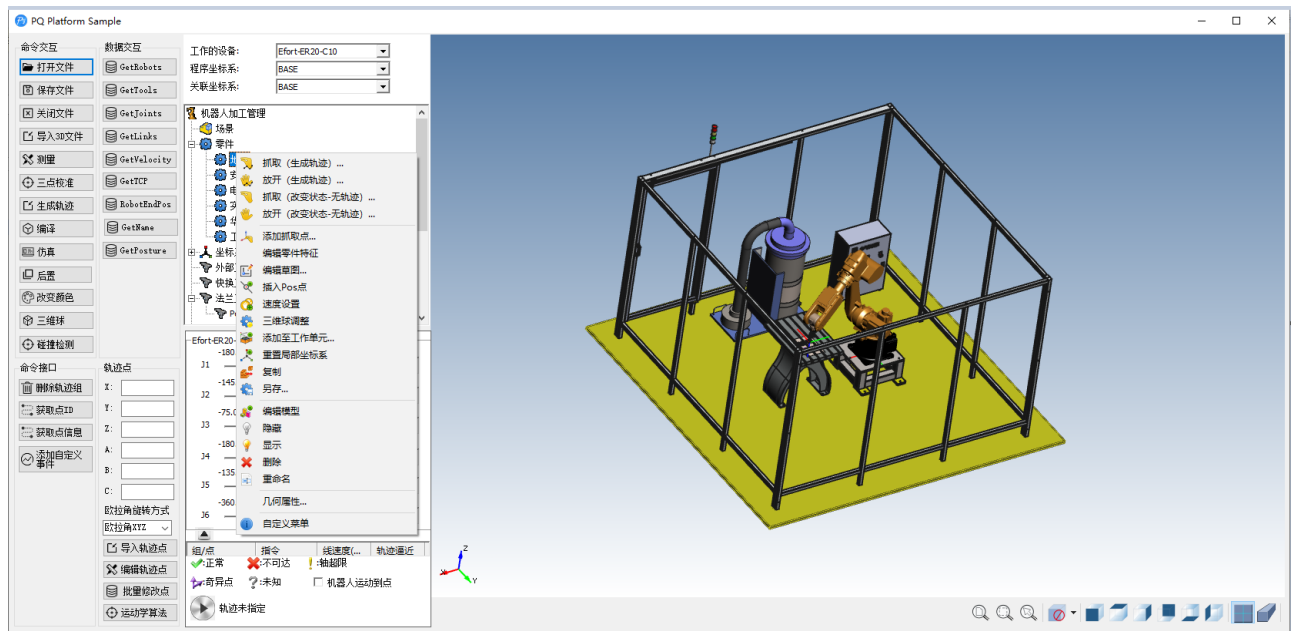
3. PQ Kit 自定义菜单

PQ Ki 支持自定义菜单，即用户可以通过自定义菜单功能对 PQ Kit 的右键菜单进行自行裁剪，以满足不同用户的定制化需求。

自定义菜单的用户行为可以发生在 PQ Kit SDK 开发者层面，即开发者定制



菜单后，发布自己的产品，PQ Kit 的最终用户能看到的就是裁剪后的菜单。也可以发生在 PQ Kit 的最终用户层面，即开发者不裁剪菜单，最终用户去裁剪菜单。



在自定义菜单功能中，可以添加某个命令（某一类型对象支持的命令）至菜单中，也可以将某个命令移出菜单。可以新建分组并添加命令、也可以将分组内的某个命令移出。可以上下移动命令改变其在菜单中的顺序。

3.1 自定义菜单界面介绍

自定义菜单中，顶部下拉列表显示所有支持更改的菜单，并会自动匹配用户选中的菜单。中间左侧列表控件分别显示选中菜单下所有的菜单命令。中间右侧列表控件显示选中菜单当前的组织结构。例如，用户在机器人右键菜单中单击自定义菜单，那么顶部下拉列表会自动选中机器人菜单，中间左侧列表控件显示所有机器人菜单命令，右侧列表控件显示当前机器人右键菜单组织结构。



3. 2 添加、删除

3.2.1 删除

右侧列表选中，单击删除按钮，即可将该命令从现有菜单中删除。

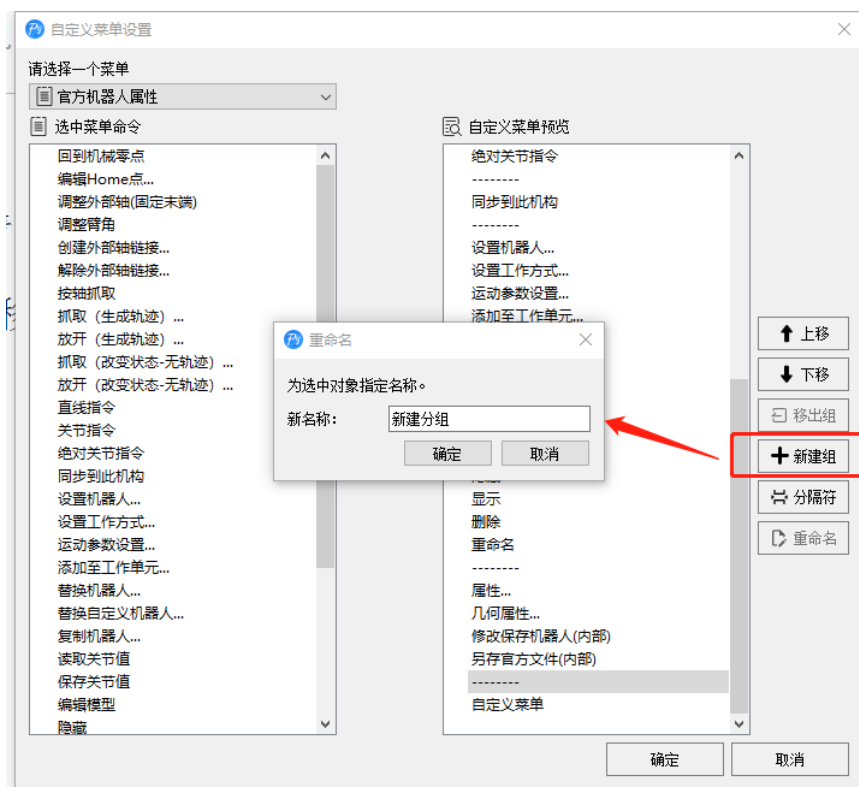
3.2.2 添加

左侧选中某个命令，单击添加按钮，即可将该命令加入现有菜单。如添加时右侧选中了某个命令，则将左侧选中命令加到右侧选中命令之后，如添加时右侧选中了某个分组，则将左侧选中命令加到右侧选中分组中。

3.3 上移下移

右侧选中了某个命令或者分组，单击上移、下移按钮，可更改选中命令或者分组在现有菜单中的位置。

3.4 分组



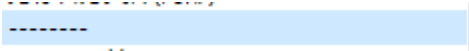
单击新建组按钮可创建新的命令分组，可以在创建时为分组指定名称，也可以随时选中分组进行重命名操作。

分组新建后，右侧列表选中该分组，从左侧列表中选择命令，单击添加按钮，即可将该命令加入新建分组中。

选中右侧列表分组下的命令，单击移出组按钮，即可将选中命令移出该分组。移出的命令将放置于顶层菜单末尾。

3.5 分隔符



即为分隔符，单击分隔符按钮，可向现有菜单中添加一条分隔符。如添加时右侧选中了某个命令，则将分隔符加到右侧选中命令之后，如添加时右侧选中了某个分组，则将分隔符加到右侧选中分组中。