

2022-04-04

...

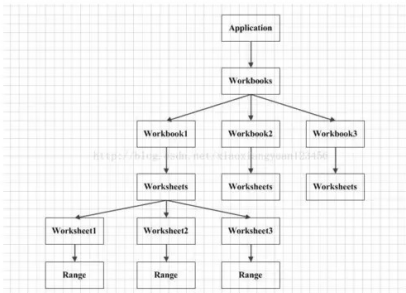
NX二次开发API里没有对EXCAL读写操作的相关函数。市面上有很多方法去实现，比如UFUN调用KF，ODBC，OLE(COM组件)等等。这里我是用的OLE(COM组件)方式去做的，这种在VC上创建的方法，无论C++还是C#还是VB方式思路都是一样的。先介绍用MFC去做，然后在写一篇博客介绍怎么在NX的二次开发的向导模板里去做。

NX二次开发-基于NX开发向导模板的NX对Excel读写操作(OLE方式(COM组件))https://ufun-nxopen.blog.csdn.net/article/details/88922238

(再看这篇博客前，先把当前博客看完。)

版本：VS2013 office2016

1.原理



Excel操作原理图

其中Application是Excel应用程序，每个Application都有一个工作簿容器Workbooks，工作簿容器中存放着所有的工作簿Workbook，我们平常打开的Excel都是一个工作簿；而每个工作簿中都有一个工作表容器Worksheets，工作表容器中存放着所有的工作表Worksheet；每个工作表中都许多单元Range，往Excel单元格中写入内容的过程就是这样一层层往下遍历。

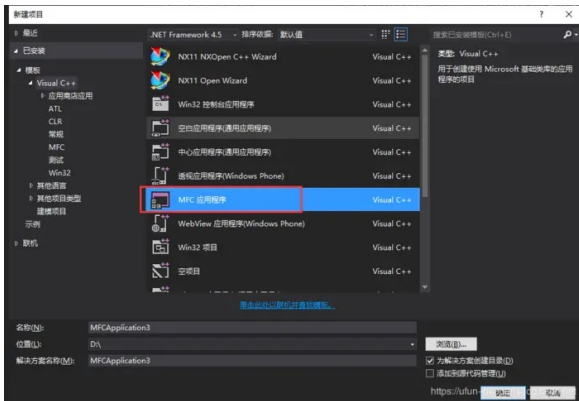
图片转载自https://blog.csdn.net/xiaoxiangyuan123456/article/details/70941588，感谢博主。

• EXCAL操作(一)打开写入

参考例子：https://blog.csdn.net/ywx123_/article/details/77074038

这是另一个CSDN博主写的，我也是参考他的例子去做的，再此感谢这位博主。

新建一个MFC项目



点击下一步



选择基于对话框



OLE(COM组件)方式写Excel

2021-07-13

NX二次开发-UFUN读取显示在NX交互界面

2021-11-29

UG/NX 二次开发

2021-08-22

用C#做UG NX 二次开发 / NX Secondary

2021-06-13

NX二次开发-NXOpenC++属性操作

2022-01-30

NX二次开发-NXOpenC++邮件操作

2022-03-07

VC用OLE方式读写Excel

2022-01-10

NX二次开发 克隆

2021-10-22

MFC中对Excel进行访问操作(OLE方式)

2022-01-01

相似解决方案

Nx Storybook Webpack"开发"模式

2021-03-09

扩展 NX 生成组件

2021-09-08

以正确的方式打开/关闭tson xavier nx

2021-07-07

删除基于 Environment Nx Angular 的代码2021-09-29

vue 作为 nx 插件并使用环境变量 (nx-plus2022-06-22

如何使用 NX Open 获取 NX 中所有开放部

2021-10-27

在 Angular Nrwl Nx 中创建组件作为库

2019-05-21

nx 导航可以返回多个操作吗？

2021-06-08

重复 N1:Nx 行 Y1:Yx 次？

2019-08-02

Nx 工作区:Angular CLI 在迁移到 NX 后找

2022-01-28

最近更新的文章/教程

更多

Harbor企业级私服Docker镜像仓库搭建及

2022-07-20

10 Dubbo 配置实战

2022-07-20

深入浅出 Yolo 系列之 YOLOv7 基础网络结

2022-07-20

ORM框架介绍——什么是ORM框架？

2022-07-20

IDEA中web项目打成war包并在本地tomcat

2022-07-20

网页数据抓取公式之京东商品数据抓取

2022-07-20

常用的函数式接口 Predicate接口 默认方

2022-07-20

MySQL 临时表用表锁,临时表用行锁？

2022-07-20

java基础语法

2022-07-20

暑假打工2个月,让我明白了Keepalived

2022-07-20

热门标签

Java Python linux javascript Mysql

C# Docker 算法 SpringBoot Redis

前端 spring Vue .net core

设计模式 .net kubernetes 数据库

c++ js 微服务 大数据 机器学习

数据结构 Android 程序员 面试 Go

JVM PHP ASP.net core git CSS

后端 k8s mybatis 爬虫 Nginx

多线程 Django React Spring Boot

golang 云计算 容器 devops 分布式

深度学习 云原生 架构

常用小工具

更多

JSON格式美化工具

在线XML转JSON/JSON转XML工具

JSON格式化编辑和美化工具

密码安全性在线检测

在线计算器

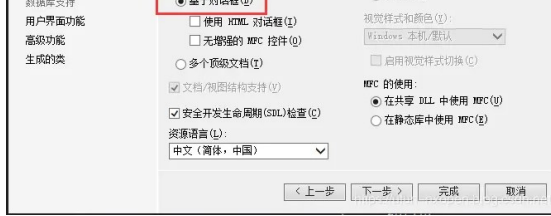
在线高级科学计算器

贷款计算器/房贷计算器

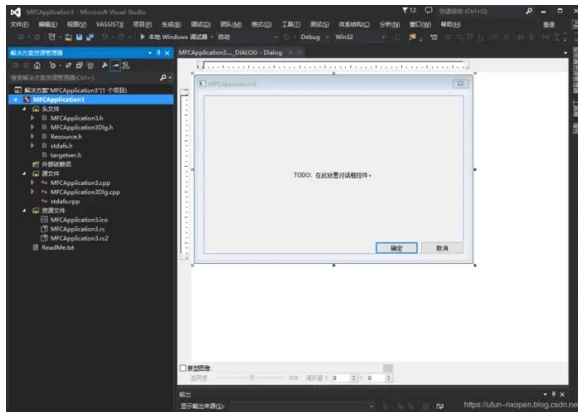
在线RGB、HEX颜色代码生成器

在线WEB安全色查询工具

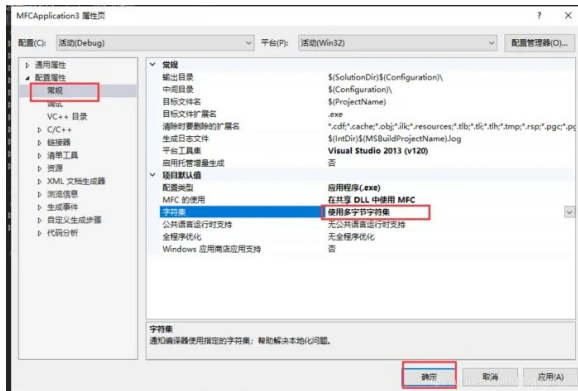
网页颜色搭配表及颜色搭配技巧



直接点完成，进来后如下图

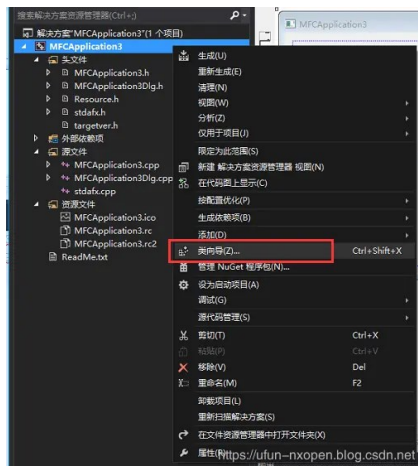


先点项目，右键属性，更改为多字节。



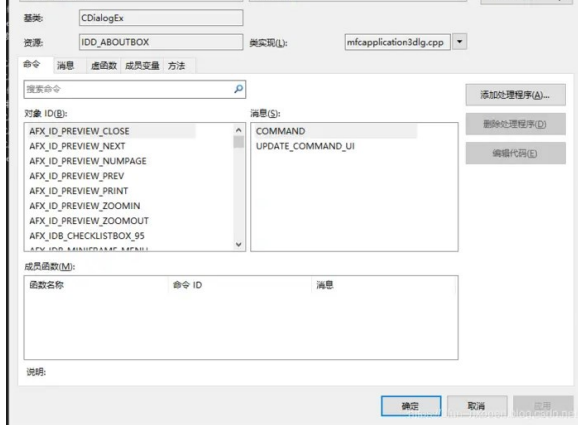
先什么都不动，编译一下代码。默认看能不能通过。

然后点项目，右键类向导。

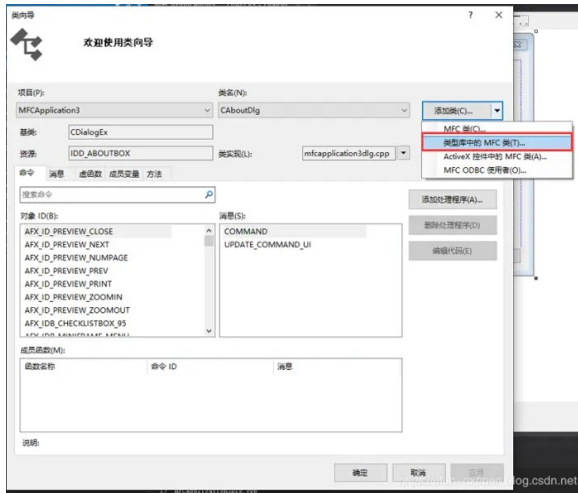


弹出窗口如下





点击添加类-类型库中的MFC类



弹出如下

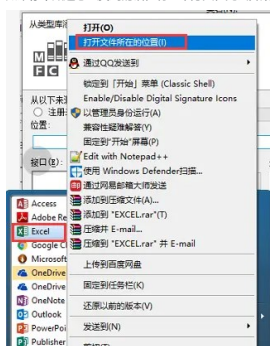


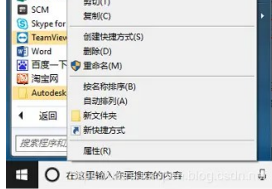
我们切换到文件，点击添加。



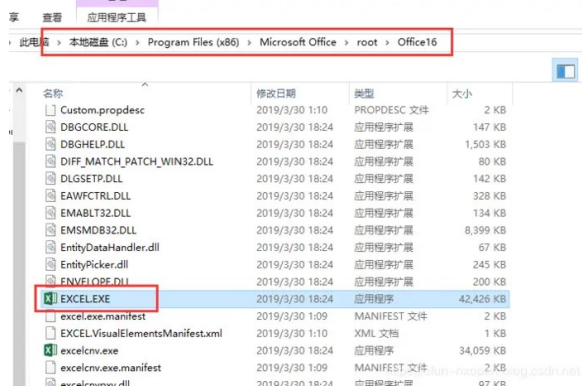
此处添加要找到的文件就是你电脑上装的EXCAL.exe这个程序文件。

如果你不知道该怎么找到的话，有一个方法，先去开始菜单找到你的EXCAL快捷方式，然后在点右键，打开文件所在的位置。





这样就找到了。



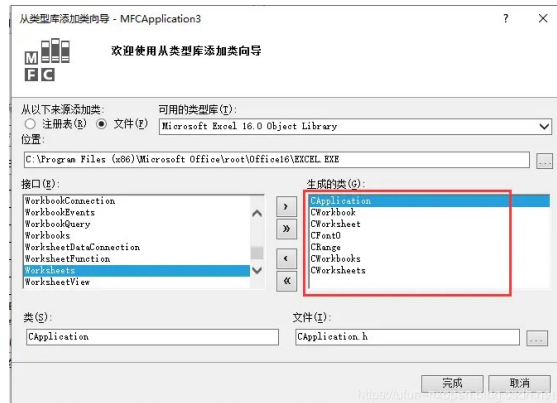
下面在MFC项目里面，我们选中这个文件。

添加进来之后，如图所示。

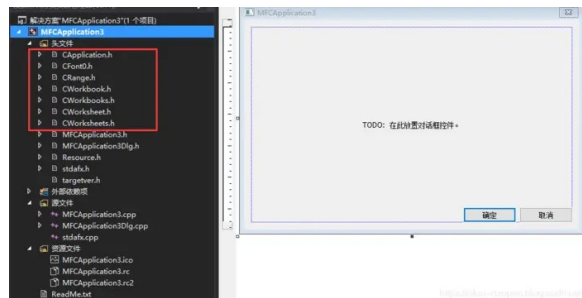


左侧为EXCAL给出的接口，我们来选择一些需要的添加进来。

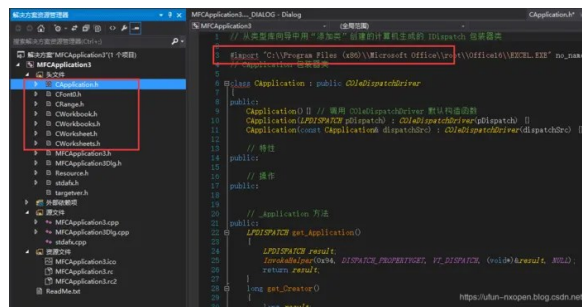
我们就先添加这七个，后续有其他需求在添加其他的。（先在左侧接口里选一个，点中间的>就能添加到右侧里生成的类里，不想要就在点中间的<撤回回去。）

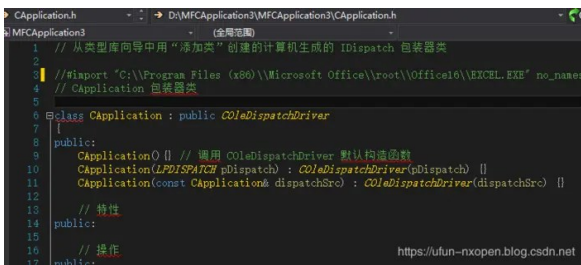


然后我们点完成，点确定。就看到头文件自动添加进来了。

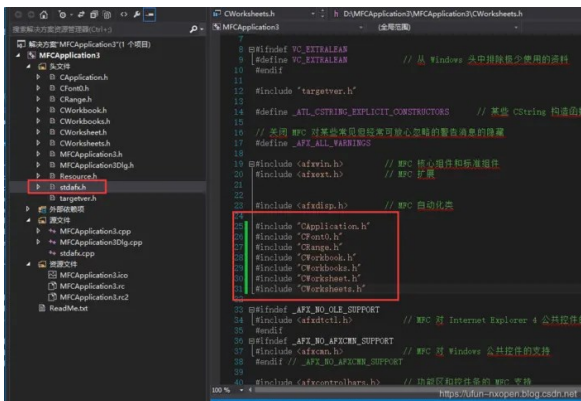


我们点击新增进来的EXCAL每个头文件，都把第一行的那句注释掉。

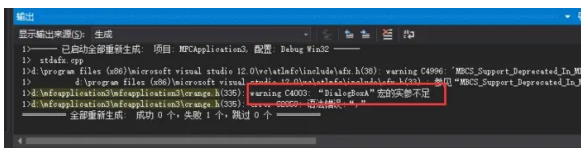




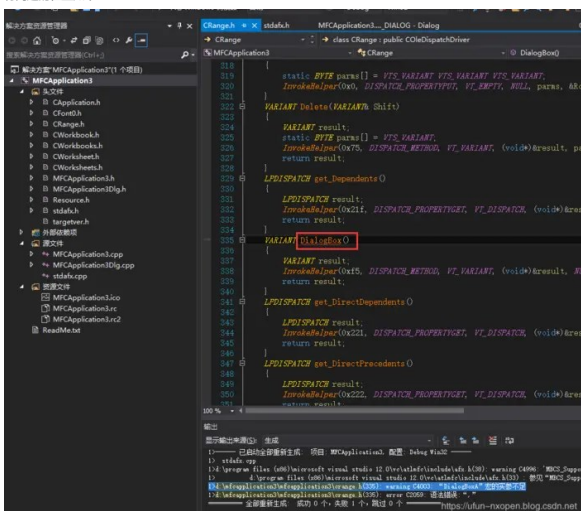
都注释完之后，我们在#include "stdafx.h"里添加EXCAL头文件。



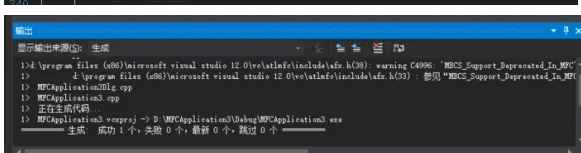
然后编译代码，出现如下报错问题。这时我们双击这个错误，跳到对应的代码那里。



有问题的位置如下



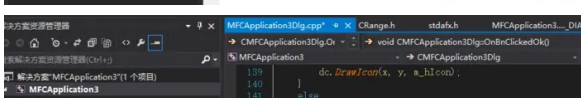
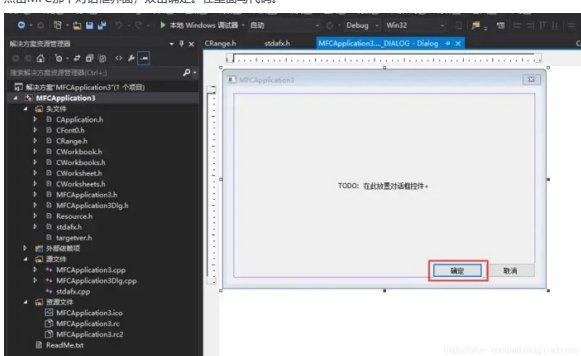
我们需要在DialogBox前面加上一个下划线。再去编译就通过了。别问我为什么，我答不上来，你照着搞就是了，别人就是这么搞的。

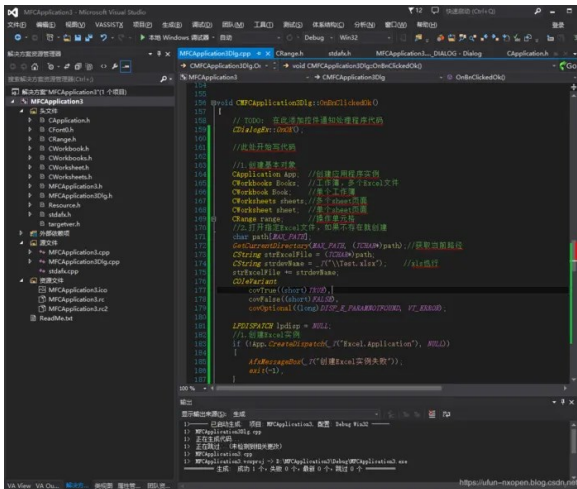
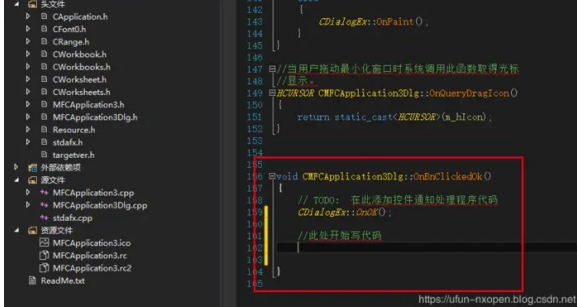


好的，到这里我们这个项目环境就算搭建完成了。

下面可以写代码了。

点击MFC那个对话框界面，双击确定，在里面写代码。





代码如下：打开EXCAL，写入内容。

```
//1. 创建基本对象
CApplication App; //创建应用程序实例

CWorkbooks Books; //工作簿，多个Excel文件

CWorkbook Book; //单个工作簿

CWorksheets sheets; //多个sheet页面

CWorksheet sheet; //单个sheet页面

CRange range; //操作单元格

//2. 打开指定Excel文件，如果不存在就创建

char path[MAX_PATH];

GetCurrentDirectory(MAX_PATH, (TCHAR*)path); //获取当前路径

CString strExcelFile = (TCHAR*)path;

CString strdevName = _T("\\Test.xlsx"); //xls也行

strExcelFile += strdevName;

COleVariant

covTrue((short)TRUE),

covFalse((short)FALSE),

covOptional((long)DISP_E_PARAMNOTFOUND, VT_ERROR);

LPDISPATCH lpdisp = NULL;

//1. 创建Excel实例

if (!App.CreateDispatch(_T("Excel.Application"), NULL))

{

    AfxMessageBox(_T("创建Excel实例失败"));

    exit(-1);

}

else

{

    AfxMessageBox(_T("创建成功"));

}

App.put_Visible(TRUE); //打开Excel

App.put_UserControl(FALSE);

//2. 得到workbooks容器

Books.AttachDispatch(App.get_Workbooks());

Book.AttachDispatch(Books.Add(covOptional));

sheets.AttachDispatch(Book.get_Worksheets());

sheet.AttachDispatch(sheets.get_Item(COleVariant((short)1))); //获取sheet1

sheet.put_Name(_T("TestName")); //设置sheet1名字

//3. 加载要合并的单元格

range.AttachDispatch(sheet.get_Range(COleVariant(_T("B2")), COleVariant(_T("B2")), TRUE);

range.Merge(COleVariant((long)0)); //合并单元格

//4. 设置表格内容

range.AttachDispatch(sheet.get_Cells(), TRUE); //加载所有单元格

range.put_Item(COleVariant((long)2), COleVariant((long)2), COleVariant(_T("电气工程及其自动化课程统计")));

range.put_Item(COleVariant((long)3), COleVariant((long)2), COleVariant(_T("课程名称")));

range.put_Item(COleVariant((long)3), COleVariant((long)3), COleVariant(_T("课时")));

range.put_Item(COleVariant((long)3), COleVariant((long)4), COleVariant(_T("难度")));

range.put_Item(COleVariant((long)3), COleVariant((long)5), COleVariant(_T("教学方式")));

range.put_Item(COleVariant((long)4), COleVariant((long)2), COleVariant(_T("电磁场")));

range.put_Item(COleVariant((long)4), COleVariant((long)3), COleVariant(_T("30")));

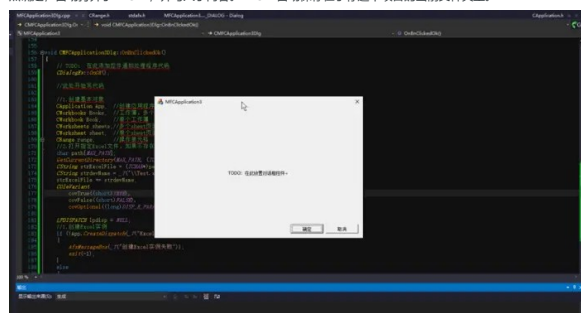
range.put_Item(COleVariant((long)4), COleVariant((long)4), COleVariant(_T("变电站")));

range.put_Item(COleVariant((long)4), COleVariant((long)5), COleVariant(_T("老师讲课"));
```

编译后，我们点调试-开始执行。



点确定，自动打开了EXCAL，并写入了内容。EXCAL自动保存在了你这个项目的当前文件夹里。



就写到这里了，好多我也没有用过，回头用到了再去研究。其他的相关内容大家参考下面这些大神们的博客吧。

参考资料：

- 一个封装的实用的EXCEL类 <https://blog.csdn.net/gyssoft/article/details/1592104>
 - vs对Excel读写操作 https://blog.csdn.net/qq_16334327/article/details/81353248
 - 在 VS2008 下操作 Excel 的方法总结 <https://blog.csdn.net/DavidHsing/article/details/4231592>
 - VS2010 对Excel读写操作 https://blog.csdn.net/ywx123_/article/details/77074038
 - VS2010 MFC Excel （1） <https://www.cnblogs.com/zztong/p/6695283.html>
 - VS2010 MFC 读取Excel （2） <https://www.cnblogs.com/zztong/p/6695284.html>
 - VS2010对Excel2010进行操作 <https://blog.csdn.net/xiaoxiangyuan123456/article/details/70941588>
 - （转） VS2010 MFC 操作Excel(读写)<https://blog.csdn.net/zhaodan19861107/article/details/80675888>
 - VS2010 MFC读取Excel文件中数据 https://blog.csdn.net/V10_x/article/details/78464453
 - MFC vs2012 Office2013 读写excel文件 （OLE/COM） https://blog.csdn.net/baidu_37503452/article/details/72842573
 - vs创建控制台程序使用C++读写excel文件 （ODBC方式） <https://www.cnblogs.com/renjiashuo/p/7545784.html>
 - VC++遍历Excel<https://blog.csdn.net/z3312656/article/details/8282194>
- 感谢各位大神无私的奉献。

原文链接：<https://www.cnblogs.com/nxopen2018/p/10981416.html>

分类：

技术点：

相关文章：

NX二次开发之User Tools	2021-08-03
NX二次开发-NXOpen::CoordinateSystemCollection Class Reference	2021-10-07
用Java做UG NX 二次开发	2022-01-27
NX二次开发-NXOpen::WCS Class Reference	2022-03-31
.net做UG NX二次开发 (VB.net) / NX Secondry Dev. with .net using VB.net Code	2021-08-16
基于VB.NET的NX UG软件二次开发基础	2021-11-24
NX二次开发 开发总结	2021-07-13
NX Open for Java[ug-nx-java-二次开发-新手] 学习手册(一)	2021-12-30
NX二次开发, 对象上色	2021-11-21
NX二次开发--NX中自身的位图查找方法	2021-08-20
NX二次开发-NX11创建表达式组NXOpen::ExpressionGroup	2022-01-20
NX二次开发-NXOpen::Drawings::DrawingSheet Class Reference	2021-12-13