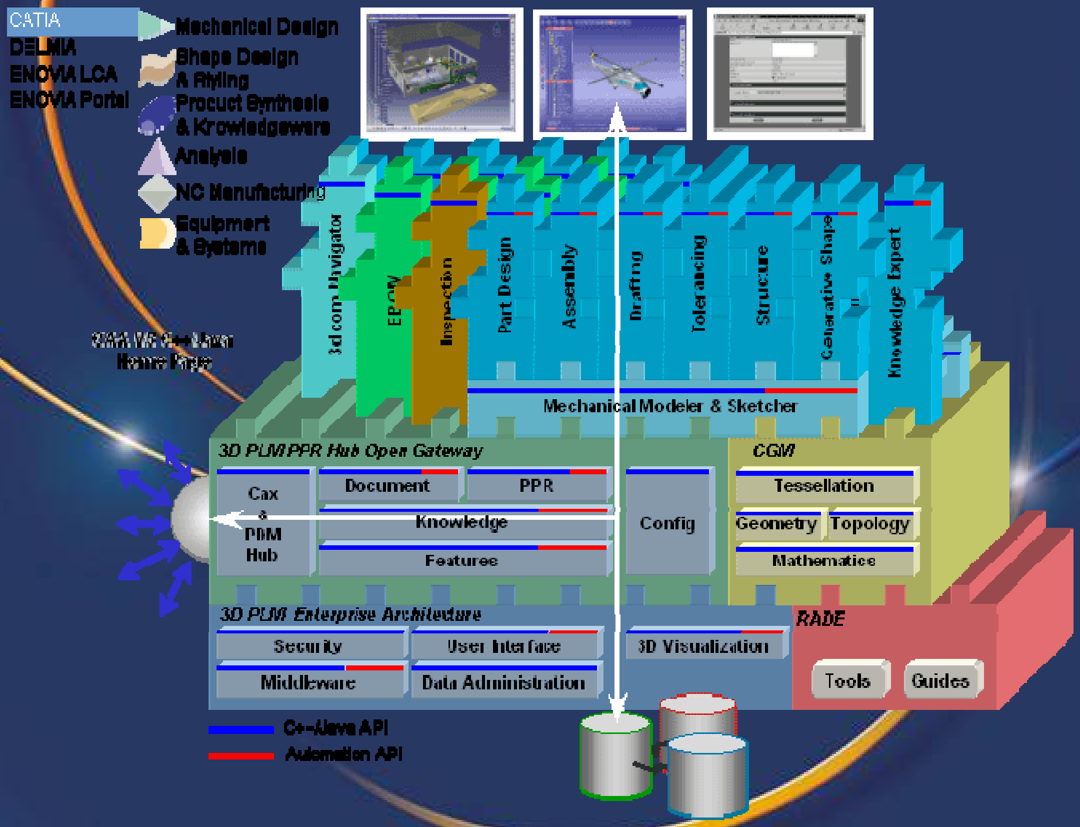
Jan. 2009

CAA V5 – 中级培训

– 自定义特征



鲁军

上海江达科技发展有限公司

DASSAULT SYSTEMES

内容介绍

- 建立**Catalog**
- 建立**Interface**
- 建立**Implementation**
- 建立用户交互（**UI**）

内容介绍

- 建立Catalog
- 建立Interface
- 建立Implementation
- 建立用户交互（UI）

建立Catalog

- 作用
  - 存放自定义特征的参考（Reference）
  - 存放自定义特征参考的属性
- 文件
  - 文件名：${CatalogFileName}.CATfct
  - 存放目录：${Workspace} + ${Framework} + CNext + resources + graphic

建立Catalog步骤

- 建立Workspace
  - Name：JDCATNewFeaWsp
- 建立Framework
  - Name：JDCATNewFeaCatalogFrm
  - Type: Implementation
  - Function: Development
- 建立Module
  - Name：JDCATCreateCatalog
  - Build mode：Executable
- 建立cpp文件
  - Name：JDCATCreateCatalog.cpp
  - Directory: ${Workspace} + ${Framework} + ${Module}.m + src

建立Catalog步骤

- cpp文件中main函数
  - 建立一个CATIA对话

    ```
    char *pSessionName = “MyCreateCatalogSession”;
    CATSession *pSession = NULL;
    HRESULT rc = ::Create_Session(pSessionName,pSession);
    ```
  - 建立Catalog

    ```
    CATUnicodeString CatalogStorageName(“JDCATNewFeat.CATfct”);
    CATICatalog *piCatalog = NULL;
    rc = ::CreateCatalog(& CatalogStorageName,&piCatalog);
    ```
  - 设定访问该Catalog文件的参数

    ```
    CATUnicodeString ClientId = “JiangDaNewFeat”;
    rc = piCatalog->SetClientId(&ClientId);
    ```
  - 建立自定义特征对象

    ```
    CATUnicodeString StartUpType = “JDExtendPT”;
    CATUnicodeString CatalogName = “CATHybridShape”;
    CATUnicodeString superTypeName = “GSMGeom”
    CATISpecObject *piSpecOnStartUp = NULL;
    rc = ::CATOsmSUFactory(&piSpecOnStartUp,&StartUpType,
                                  piCatalog,&SuperTypeName,
                                  &CatalogName,TRUE,TRUE);
    ```

建立Catalog步骤

- cpp文件中main函数
  - 建立属性

    ```
    CATUnicodeString AttName1(“px”);
    CATISpecAttribute *piAtt1 = NULL;
    piAtt1 = piSpecOnStartUp->AddAttribute(AttName1,tk_double,sp_IN);
    if( piAtt1 != NULL ) {
      piAtt1->Release();
      piAtt1 = NULL;
    }
    else {
      cout << “=> Create attribute error !” << endl;
      return;
    }
    ```

    用以上方法建立 py,pz 属性
  - 保存Catalog文件

    ```
    rc = ::SaveCatalog(&piCatalog,&CatalogStorageName);
    ```
  - 关闭CATIA对话

    ```
    rc = ::Delete_Session(pSessionName);
    ```

建立Catalog步骤

- 定义"Locate Prerequisite Workspaces"
- 定义所需的Framework：IdentityCard.h
- 定义所需的Module：Imakefile.mk
- 编译并建立运行环境：mkmk - mkrtv
- 运行：JDCATCreateCatalog.exe
- 拷贝JDCATNewFeat.CATfct文件到

  ${Workspace}\${Framework} \CNext\resources\graphic目录

内容介绍

- 建立Catalog
- 建立Interface
- 建立Implementation
- 建立用户交互（UI）

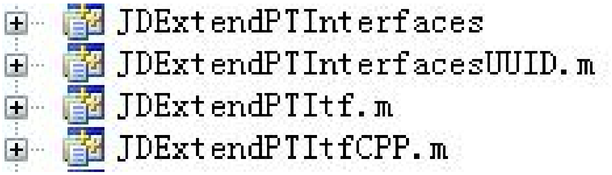
建立Interface

- 作用
  - 建立自定义特征的接口
  - 访问自定义特征属性的接口

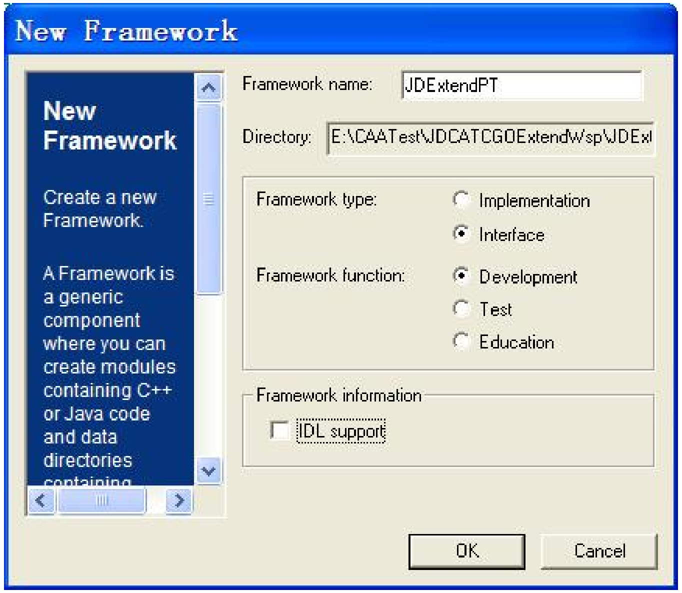
建立Interface步骤

- 建立Interface的Framework
  - Name：JDExtendPT
  - Type：Interface
  - Function：Development
  - No IDL Support

  - 刷新全部工程



```
⊞ JDExtendPTInterfaces
⊞ JDExtendPTInterfacesUUID.m
⊞ JDExtendPTItf.m
⊞ JDExtendPTItfCPP.m
```



New Framework

New Framework

Create a new Framework.

A Framework is a generic component where you can create modules containing C++ or Java code and data directories containing

Framework name: JDExtendPT

Directory: E:\CAATest\JDCATCGOExtendWsp\JDExt

Framework type: ○ Implementation ● Interface

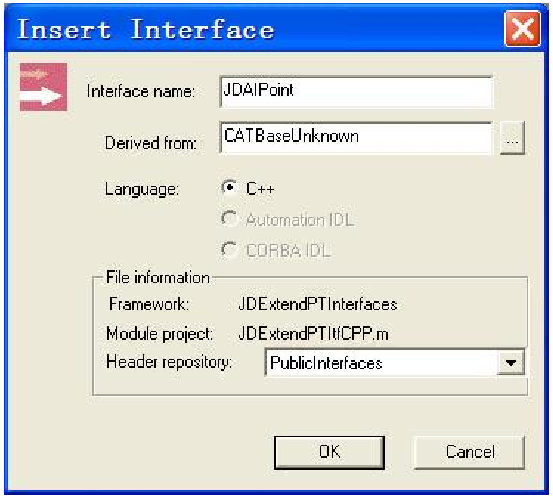
Framework function: ● Development ○ Test ○ Education

Framework information

☐ IDL support

建立Interface

- 建立自定义特征的接口
  - 设定JDExtendPTInterfaces为启动项目
  - Name：JDAIPoint
  - Derived from：CATBaseUnknown
  - Header repository：PublicInterfaces



Insert Interface

Interface name: JDAIPoint

Derived from: CATBaseUnknown

Language: ● C++ ○ Automation IDL ○ CORBA IDL

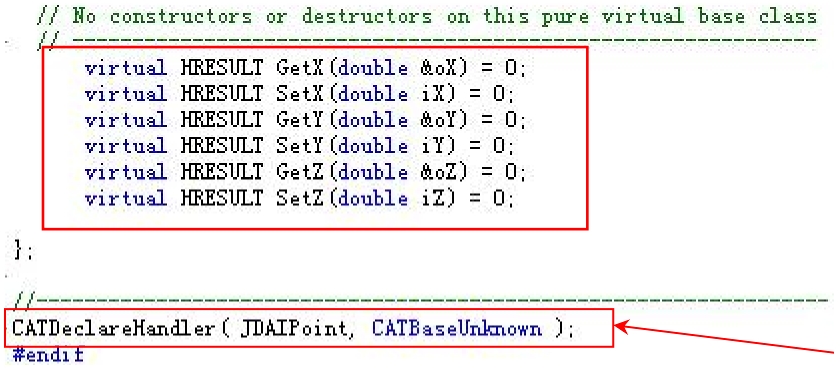
File information

Framework: JDExtendPTInterfaces

Module project: JDExtendPTItfCPP.m

Header repository: PublicInterfaces

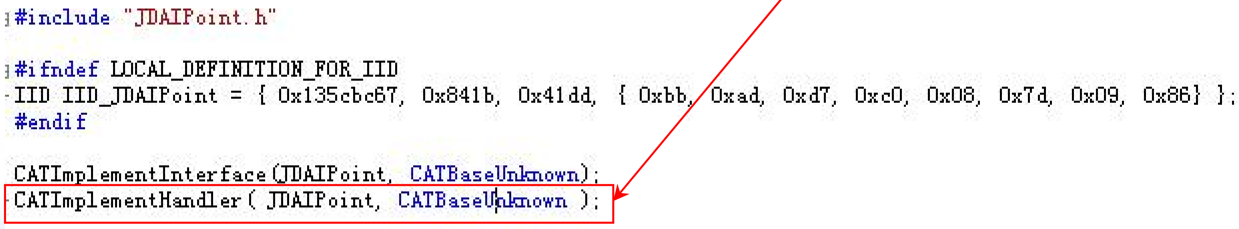
  - 在JDAIPoint.h文件中，定义访问自定义特征属性的方法



```
// No constructors or destructors on this pure virtual base class
// --------------------------------------------------------------
   virtual HRESULT GetX(double &oX) = 0;
   virtual HRESULT SetX(double iX) = 0;
   virtual HRESULT GetY(double &oY) = 0;
   virtual HRESULT SetY(double iY) = 0;
   virtual HRESULT GetZ(double &oZ) = 0;
   virtual HRESULT SetZ(double iZ) = 0;

};

//------------------------------------------------------------------
CATDeclareHandler( JDAIPoint, CATBaseUnknown );
#endif
```

使JDAIPoint_var有效

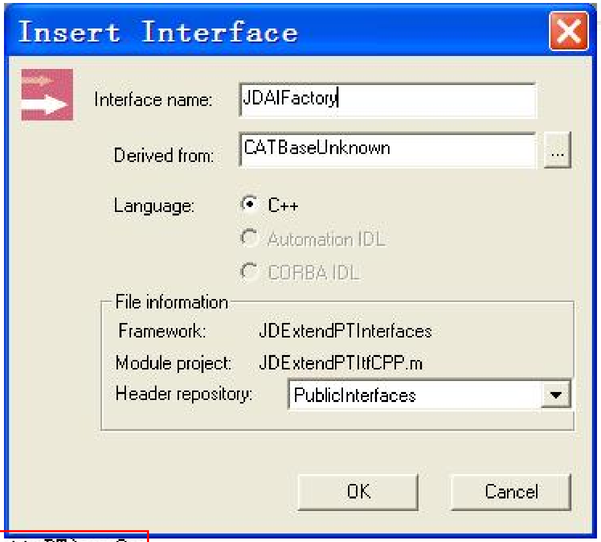
  - 在JDAIPoint.cpp文件中，定义：



```
#include "JDAIPoint.h"

#ifndef LOCAL_DEFINITION_FOR_IID
IID IID_JDAIPoint = { 0x135cbc67, 0x841b, 0x41dd, { 0xbb, 0xad, 0xd7, 0xc0, 0x08, 0x7d, 0x09, 0x86} };
#endif

CATImplementInterface(JDAIPoint, CATBaseUnknown);
CATImplementHandler( JDAIPoint, CATBaseUnknown );
```

建立Interface

- 建立创建自定义特征的Factory接口
  - 设定JDExtendPTInterfaces为启动项目
  - Name：JDAIFactory
  - Derived from：CATBaseUnknown
  - Header repository：PublicInterfaces



Insert Interface

Interface name: JDAIFactory

Derived from: CATBaseUnknown

Language: ● C++ ○ Automation IDL ○ CORBA IDL

File information

Framework: JDExtendPTInterfaces

Module project: JDExtendPTItfCPP.m

Header repository: PublicInterfaces

  - 在JDAIFactory.h文件中，定义创建自定义特征的方法

```
  // No constructors or destructors on this pure virtual base class
  // --------------------------------------------------------------
     virtual HRESULT CreateJDAPoint(double iX,double iY,double iZ,JDAIPoint  **oPT) = 0;
};

//------------------------------------------------------------------
CATDeclareHandler( JDAIFactory, CATBaseUnknown );
#endif
```

  - 在JDAIPoint.cpp文件中，

```
#include "JDAIFactory.h"

#ifndef LOCAL_DEFINITION_FOR_IID
IID IID_JDAIFactory = { 0x81444a5b, 0xcc20, 0x4ada, { 0x96, 0x83, 0xc1, 0x53, 0x83, 0x81, 0x11, 0x0b} };
#endif

CATImplementInterface(JDAIFactory, CATBaseUnknown);
CATImplementHandler(JDAIFactory, CATBaseUnknown );
```

使JDAIFacgtory_var有效

- 编译并建立运行环境：mkmk – mkrtv

内容介绍

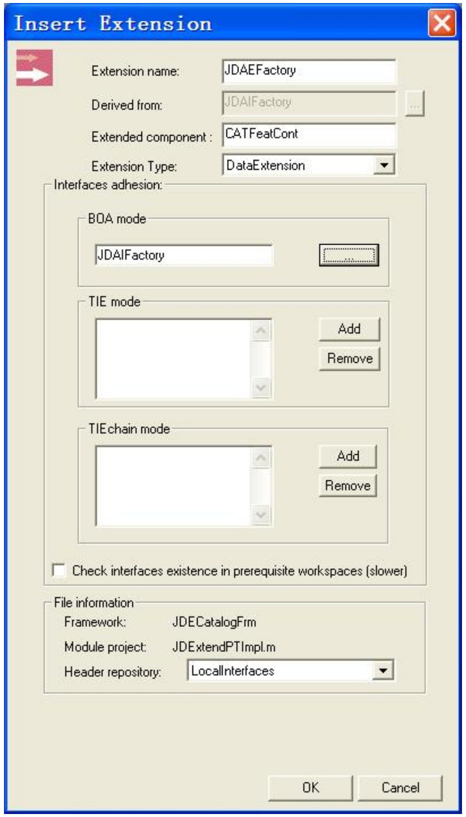
- 建立Catalog
- 建立Interface
- 建立Implementation
- 建立用户交互（UI）

建立Implementation

- 作用
  - 建立自定义特征的实现方法（Factory）
  - 建立自定义特征属性的实现方法
  - 建立自定义特征的几何构建方法（Build）
  - 建立自定义特征的编辑方法
  - 建立自定义特征图标的方法

建立Implementation步骤

- 建立Module
  - 设定JDCATNewFeaCatalogFrm为启动项目
  - Name：JDExtendPTImpl
  - Build mode：Shareed object
- 创建自定义特征的实现方法（Factory）
  - 设定JDExtendPTImpl为启动项目
  - 建立一个Component Extension
  - Estension Name：JDAEFactory
  - Extended component：CATFeatCont
  - Extension Type：DataExtension
  - BOA mode：JDAIFactory


Insert Extension

Extension name: JDAEFactory

Derived from: JDAIFactory

Extended component : CATFeatCont

Extension Type: DataExtension

Interfaces adhesion:

BOA mode: JDAIFactory

TIE mode

TIEchain mode

□ Check interfaces existence in prerequisite workspaces (slower)

File information

Framework: JDECatalogFrm

Module project: JDExtendPTImpl.m

Header repository: LocalInterfaces

建立Implementation

— JDAEFactory

- 编写JDAIFactory 中方法的具体实现：

CreateJDAPoint(double iX,double iY,double iZ,JDAIPoint **oPT)：

- 获取当前Container变量

```
CATIContainer_var spSpecContainer(this);
```

- 访问特征库

```
CATICatalog *piCatalog=NULL;
CATUnicodeString CatalogStorageName="JDCATNewFeat.CATfct";
CATUnicodeString ClientId="JiangDaNewFeat";
HRESULT rc=::AccessCatalog(&CatalogStorageName,&ClientId,
                    spSpecContainer,&piCatalog);
```

- 获取Startup

```
CATBaseUnknown *piStartup=NULL;
CATUnicodeString StartupType="JDExtendPT";
rc=piCatalog->RetrieveSU(&piStartup,&StartupType,"CATISpecObject");
piCatalog->Release(); piCatalog = NULL;
CATISpecObject *piSpecOnStartup=NULL;
rc=piStartup->QueryInterface(IID_CATISpecObject,(void **)&piSpecOnStartup);
piStartup->Release();  piStartup = NULL;
```

- 创建特征实例

```
CATISpecObject *piSpecInstance=piSpecOnStartup->Instanciate(NULL_string,
                                                    spSpecContainer);
piSpecOnStartup->Release(); piSpecOnStartup = NULL;
```

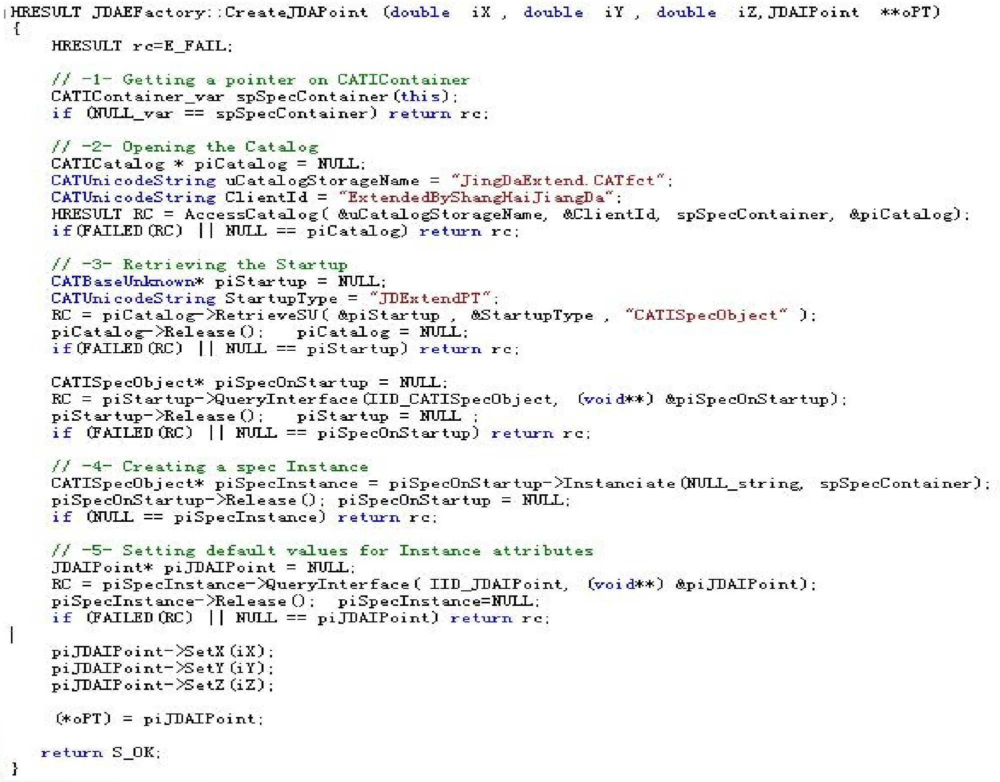
建立Implementation

— JDAEFactory

CreateJDAPoint (double iX,double iY,double iZ,JDAIPoint **oPT)：

- 设定特征实例的参数

```
JDAIPoint *piJDAIPoint=NULL;
rc = piSpecInstance->QueryInterface(IID_JDAIPoint,(void **)&piJDAIPoint);
piSpecInstance->Release();   piSpecInstance = NULL;
piJDAIPoint->SetX(iX); piJDAIPoint->SetY(iY); piJDAIPoint->SetZ(iZ);
(*oPT) = piJDAIPoint;
```

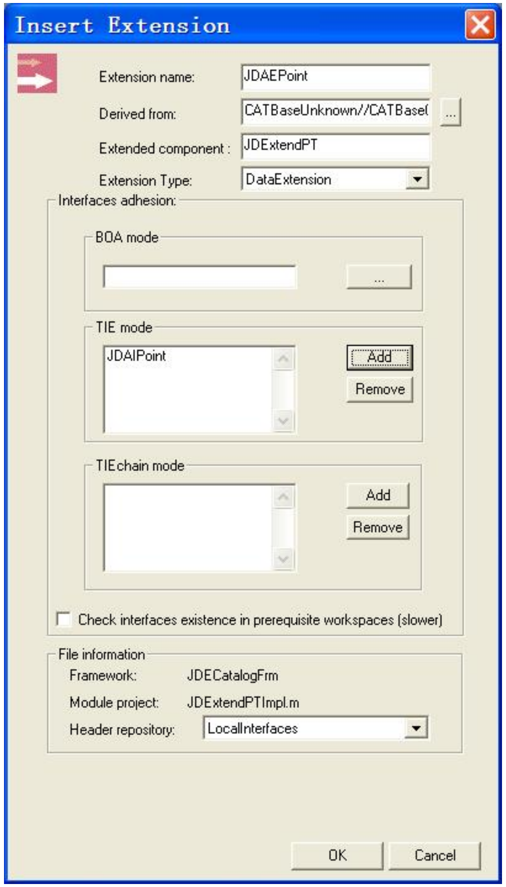


```
HRESULT JDAEFactory::CreateJDAPoint (double  iX , double  iY , double  iZ,JDAIPoint  **oPT)
{
    HRESULT rc=E_FAIL;

    // -1- Getting a pointer on CATIContainer
    CATIContainer_var spSpecContainer(this);
    if (NULL_var == spSpecContainer) return rc;

    // -2- Opening the Catalog
    CATICatalog * piCatalog = NULL;
    CATUnicodeString uCatalogStorageName = "JingDaExtend.CATfct";
    CATUnicodeString ClientId = "ExtendedByShangHaiJiangDa";
    HRESULT RC = AccessCatalog( &uCatalogStorageName, &ClientId, spSpecContainer, &piCatalog);
    if(FAILED(RC) || NULL == piCatalog) return rc;

    // -3- Retrieving the Startup
    CATBaseUnknown* piStartup = NULL;
    CATUnicodeString StartupType = "JDExtendPT";
    RC = piCatalog->RetrieveSU( &piStartup , &StartupType , "CATISpecObject" );
    piCatalog->Release();   piCatalog = NULL;
    if(FAILED(RC) || NULL == piStartup) return rc;

    CATISpecObject* piSpecOnStartup = NULL;
    RC = piStartup->QueryInterface(IID_CATISpecObject, (void**) &piSpecOnStartup);
    piStartup->Release();   piStartup = NULL ;
    if (FAILED(RC) || NULL == piSpecOnStartup) return rc;

    // -4- Creating a spec Instance
    CATISpecObject* piSpecInstance = piSpecOnStartup->Instanciate(NULL_string, spSpecContainer);
    piSpecOnStartup->Release(); piSpecOnStartup = NULL;
    if (NULL == piSpecInstance) return rc;

    // -5- Setting default values for Instance attributes
    JDAIPoint* piJDAIPoint = NULL;
    RC = piSpecInstance->QueryInterface( IID_JDAIPoint, (void**) &piJDAIPoint);
    piSpecInstance->Release();  piSpecInstance=NULL;
    if (FAILED(RC) || NULL == piJDAIPoint) return rc;

    piJDAIPoint->SetX(iX);
    piJDAIPoint->SetY(iY);
    piJDAIPoint->SetZ(iZ);

     (*oPT) = piJDAIPoint;

   return S_OK;
}
```

建立Implementation

— JDAEPoint

- 建立自定义特征属性的实现方法
  - 设定JDExtendPTImpl为启动项目
  - 建立一个Component Extension
  - Estension Name：JDAEPoint
  - Extended component：JDExtendPT
  - Extension Type：DataExtension
  - TIE mode：JDAIPoint



Extension name: JDAEPoint

Derived from: CATBaseUnknown//CATBase(

Extended component : JDExtendPT

Extension Type: DataExtension

Interfaces adhesion:

BOA mode

TIE mode

JDAIPoint

TIEchain mode

☐ Check interfaces existence in prerequisite workspaces (slower)

File information

Framework: JDECatalogFrm

Module project: JDExtendPTImpl.m

Header repository: LocalInterfaces

建立Implementation

— JDAEPoint

- 编写JDAIPoint 中方法的具体实现：

GetX(double &oX):

- 获取CATISpecAttrAccess变量

```
CATISpecAttrAccess_var spSpecAttrAccess(this);
```

- 获取CATISpecAttrKey变量

```
CATISpecAttrKey *piSpecAttrKey=spSpecAttrAccess->GetAttrKey(“px”);
```

- 获取参数值

```
double reValue=spSpecAttrAccess->GetDouble(piSpecAttrKey);
piSpecAttrKey->Release(); piSpecAttrKey = NULL;
oX = reValue;
```

SetX(double iX):

- 获取CATISpecAttrAccess变量

```
CATISpecAttrAccess_var spSpecAttrAccess(this);
```

- 获取CATISpecAttrKey变量

```
CATISpecAttrKey *piSpecAttrKey=spSpecAttrAccess->GetAttrKey(“px”);
```

- 设置参数值

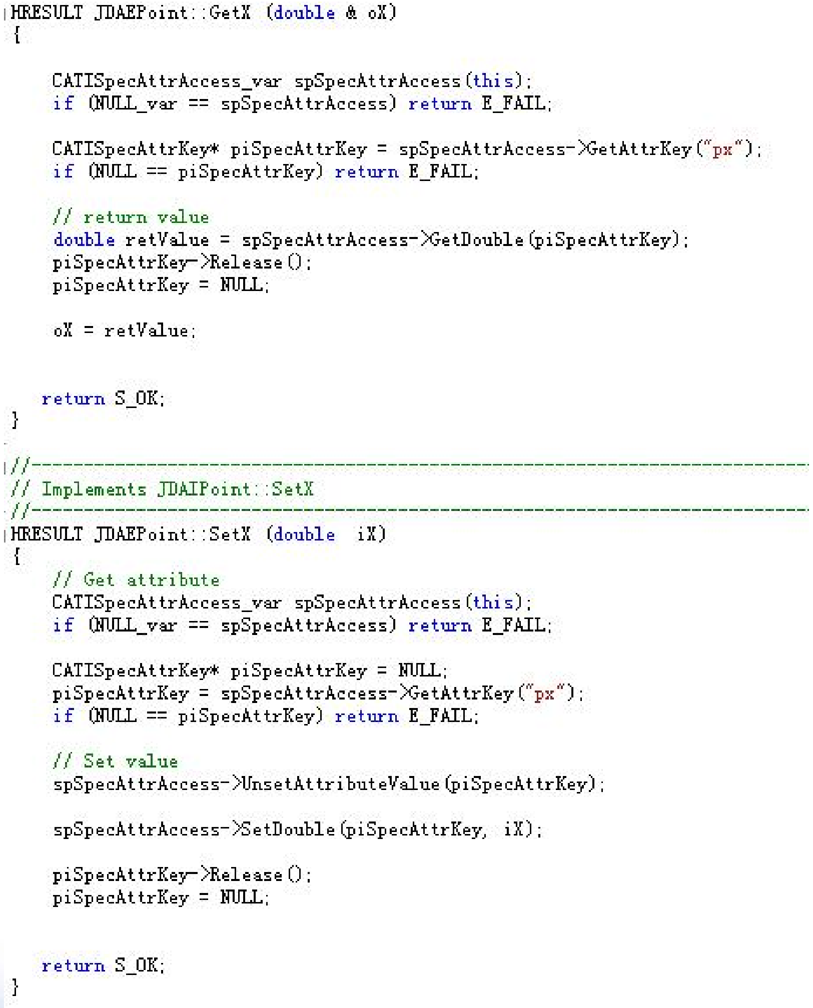
```
spSpecAttrAccess->UnsetAttributeValue(piSpecAttrKey);
spSpecAttrAccess->SetDouble(piSpecAttrKey,iX);
piSpecAttrKey->Release(); piSpecAttrKey = NULL;
```

建立Implementation

－ JDAEPoint

- 编写JDAIPoint 中方法的具体实现：

参照GetX及SetX方法，编写GetY、SetY、GetZ、SetZ方法

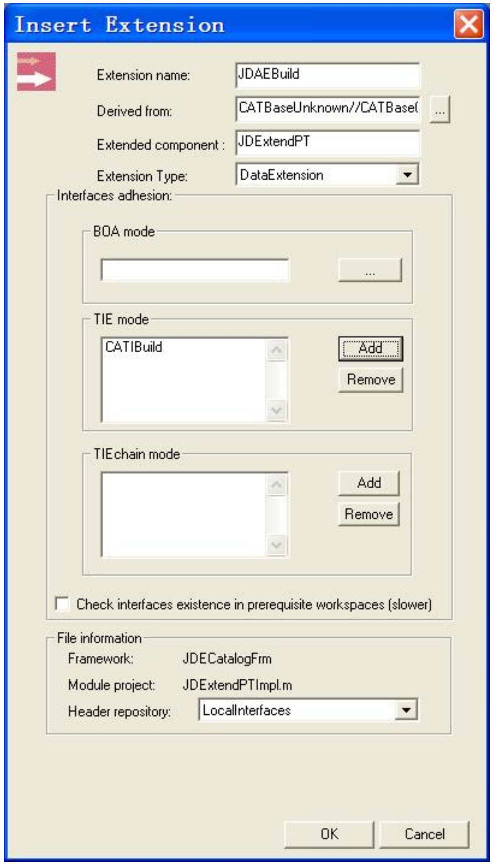


```
HRESULT JDAEPoint::GetX (double & oX)
{

    CATISpecAttrAccess_var spSpecAttrAccess(this);
    if (NULL_var == spSpecAttrAccess) return E_FAIL;

    CATISpecAttrKey* piSpecAttrKey = spSpecAttrAccess->GetAttrKey("px");
    if (NULL == piSpecAttrKey) return E_FAIL;

    // return value
    double retValue = spSpecAttrAccess->GetDouble(piSpecAttrKey);
    piSpecAttrKey->Release();
    piSpecAttrKey = NULL;

    oX = retValue;


   return S_OK;
}

//-----------------------------------------------------------------------------
// Implements JDAIPoint::SetX
//-----------------------------------------------------------------------------
HRESULT JDAEPoint::SetX (double  iX)
{
    // Get attribute
    CATISpecAttrAccess_var spSpecAttrAccess(this);
    if (NULL_var == spSpecAttrAccess) return E_FAIL;

    CATISpecAttrKey* piSpecAttrKey = NULL;
    piSpecAttrKey = spSpecAttrAccess->GetAttrKey("px");
    if (NULL == piSpecAttrKey) return E_FAIL;

    // Set value
    spSpecAttrAccess->UnsetAttributeValue(piSpecAttrKey);

    spSpecAttrAccess->SetDouble(piSpecAttrKey, iX);

    piSpecAttrKey->Release();
    piSpecAttrKey = NULL;


   return S_OK;
}
```

建立Implementation

— JDAEBuild

- 建立自定义特征的几何构建方法(Build)
  - 设定JDExtendPTImpl为启动项目
  - 建立一个Component Extension
  - Estension Name：JDAEBuild
  - Extended component：JDExtendPT
  - Extension Type：DataExtension
  - TIE mode：CATIBuild



Insert Extension

Extension name: JDAEBuild

Derived from: CATBaseUnknown//CATBase(

Extended component : JDExtendPT

Extension Type: DataExtension

Interfaces adhesion:

BOA mode

TIE mode: CATIBuild

TIEchain mode

□ Check interfaces existence in prerequisite workspaces (slower)

File information

Framework: JDECatalogFrm

Module project: JDExtendPTImpl.m

Header repository: LocalInterfaces

建立Implementation

— JDAEBuild

- 编写CATIBuild接口中方法的具体实现：

Build()：

- 删除该特征实例的所有更新错误信息

```
CATIUpdateError *piUpdateError=NULL;
HRESULT rc=this->QueryInterface(IID_CATIUpdateError,(void **)&piUpdateError);
piUpdateError->UnsetUpdateError();
CATIMfResultManagement_var spMfResultManagement(this);
if( NULL_var != spMfResultManagement ) {
   spMfResultManagement->DeleteScopeResult();
}
```

- 获取该特征实例的输入参数

```
JDAIPoint *piJDAIPoint=NULL;
rc = this->QueryInterface(IID_JDAIPoint,(void **)&piJDAIPoint);
double _x,_y,_z;
rc = piJDAIPoint->GetX(_x);
rc = piJDAIPoint->GetY(_y);
rc = piJDAIPoint->GetZ(_z);
```

- 定义一个程序报告（Procedural report）

```
CATIMfProcReport *piProcReport=NULL;
rc = piJDAIPoint->QueryInterface(IID_CATIMfProcReport,(void **)&piProcReport);
CATLISTV(CATBaseUnknown_var) ListSpec;
CATListOfCATUnicodeString ListKeys;
piProcReport->CreateProcReport(ListSpec,ListKeys);
CATCGMJournalList *pCGMJournalList=piProcReport->GetCGMJournalList();
```

建立Implementation

— JDAEBuild

- 编写CATIBuild接口中方法的具体实现：

Build()：

- 用几何/拓扑方法创建该特征实例

```
CATGeoFactory *piGeoFactory=……;
CATIDatumFactory *pDatumFactory=……;
CATSoftwareConfiguration *pConfig=new CATSoftwareConfiguration();
CATTopData *topdata=new CATTopData(pConfig,pCGMJournalList);
CATTry {
  CATBody *spBody=CATCreateTopPointXYZ(piGeoFactory,topdata,_x,_y,_z);
  piGeoFactory->Release();  piGeoFactory = NULL;
  if( spBody != NULL ) {
    piProcReport->StoreProcReport(spBody);
  } else {
    CATMfErrUpdate *pErrorNoIntersection=new CATMfErrUpdate();
    CATUnicodeString Diagnostic(“Can’t create the JDAExtendPT.”);
    pErrorNoIntersection->SetDiagnostic(1,Diagnostic);
    CATThrow(pErrorNoIntersection);
  }
}
```

建立Implementation

— JDAEBuild

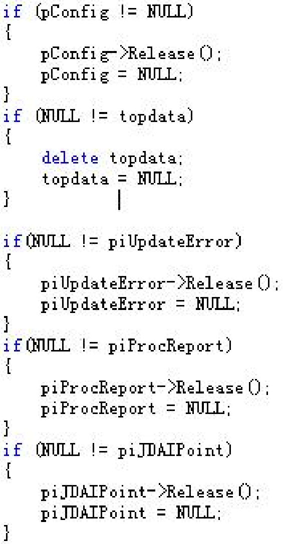
- 编写CATIBuild接口中方法的具体实现：

Build()：

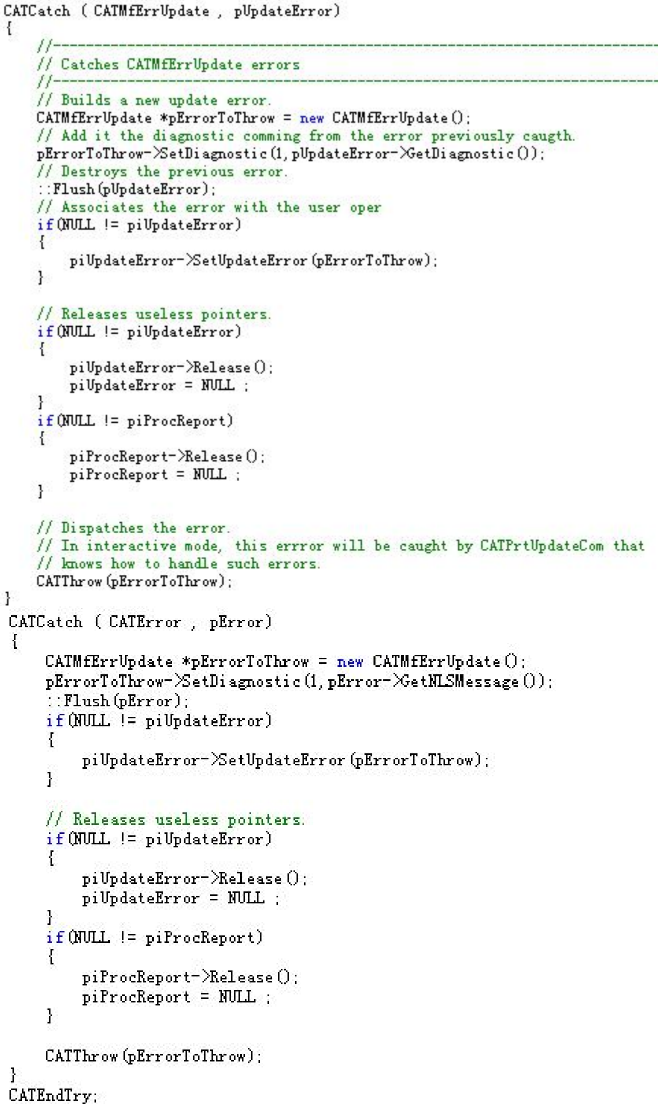
- 管理错误信息

```
CATCatch( CATMfErrUpdate,pUpdateError)
{
    ……
}
CATCatch (CATError,pError)
{
    ……
}
CATEndTry;
```

- 释放有关指针
  pConfig,
  topdata,
  piUpdateError,
  piProcReport,
  piJDAIPoint



```
if (pConfig != NULL)
{
    pConfig->Release();
    pConfig = NULL;
}
if (NULL != topdata)
{
    delete topdata;
    topdata = NULL;
}

if(NULL != piUpdateError)
{
    piUpdateError->Release();
    piUpdateError = NULL;
}
if(NULL != piProcReport)
{
    piProcReport->Release();
    piProcReport = NULL;
}
if (NULL != piJDAIPoint)
{
    piJDAIPoint->Release();
    piJDAIPoint = NULL;
}
```

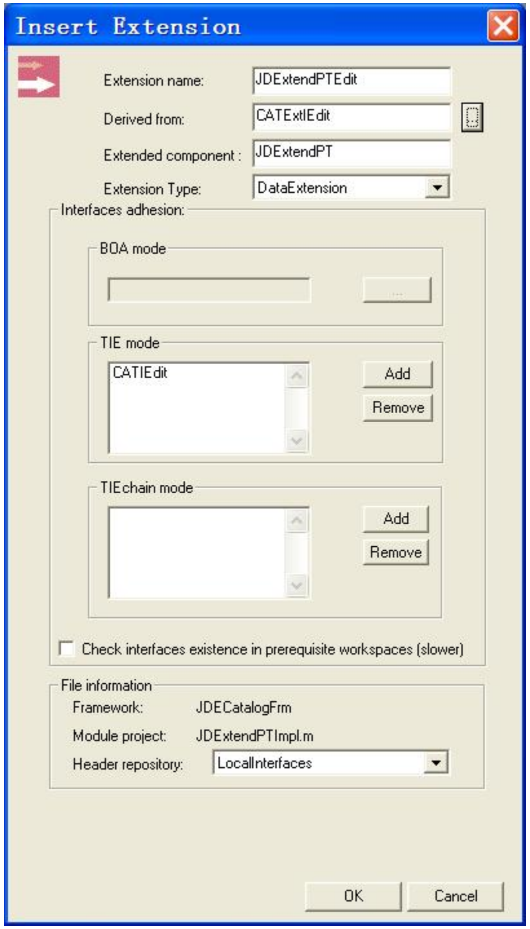


```
CATCatch ( CATMfErrUpdate , pUpdateError)
{
    //---------------------------------------------------------------------------
    // Catches CATMfErrUpdate errors
    //---------------------------------------------------------------------------
    // Builds a new update error.
    CATMfErrUpdate *pErrorToThrow = new CATMfErrUpdate();
    // Add it the diagnostic comming from the error previously caugth.
    pErrorToThrow->SetDiagnostic(1,pUpdateError->GetDiagnostic());
    // Destroys the previous error.
    ::Flush(pUpdateError);
    // Associates the error with the user oper
    if(NULL != piUpdateError)
    {
        piUpdateError->SetUpdateError(pErrorToThrow);
    }

    // Releases useless pointers.
    if(NULL != piUpdateError)
    {
        piUpdateError->Release();
        piUpdateError = NULL ;
    }
    if(NULL != piProcReport)
    {
        piProcReport->Release();
        piProcReport = NULL ;
    }

    // Dispatches the error.
    // In interactive mode, this errror will be caught by CATPrtUpdateCom that
    // knows how to handle such errors.
    CATThrow(pErrorToThrow);
}
 CATCatch ( CATError , pError)
 {
     CATMfErrUpdate *pErrorToThrow = new CATMfErrUpdate();
     pErrorToThrow->SetDiagnostic(1,pError->GetNLSMessage());
     ::Flush(pError);
     if(NULL != piUpdateError)
     {
         piUpdateError->SetUpdateError(pErrorToThrow);
     }

     // Releases useless pointers.
     if(NULL != piUpdateError)
     {
         piUpdateError->Release();
         piUpdateError = NULL ;
     }
     if(NULL != piProcReport)
     {
         piProcReport->Release();
         piProcReport = NULL ;
     }

     CATThrow(pErrorToThrow);
 }
 CATEndTry;
```

建立Implementation

— JDExtendPTEdit

- 建立自定义特征的编辑方法
  - 设定JDExtendPTImpl为启动项目
  - 建立一个Component Extension
  - Estension Name：JDAExtendPTEdit
  - Derived from:CATExtIEdit
  - Extended component：JDExtendPT
  - Extension Type：DataExtension
  - TIE mode：CATIEdit



Insert Extension

Extension name: JDExtendPTEdit

Derived from: CATExtIEdit

Extended component : JDExtendPT

Extension Type: DataExtension

Interfaces adhesion:

BOA mode

TIE mode: CATIEdit

TIEchain mode

□ Check interfaces existence in prerequisite workspaces (slower)

File information

Framework: JDECatalogFrm

Module project: JDExtendPTImpl.m

Header repository: LocalInterfaces

建立Implementation

— JDExtendPTEdit

- 编写CATIEdit接口中方法的具体实现：

Activate(CATPathElement *iPath)：

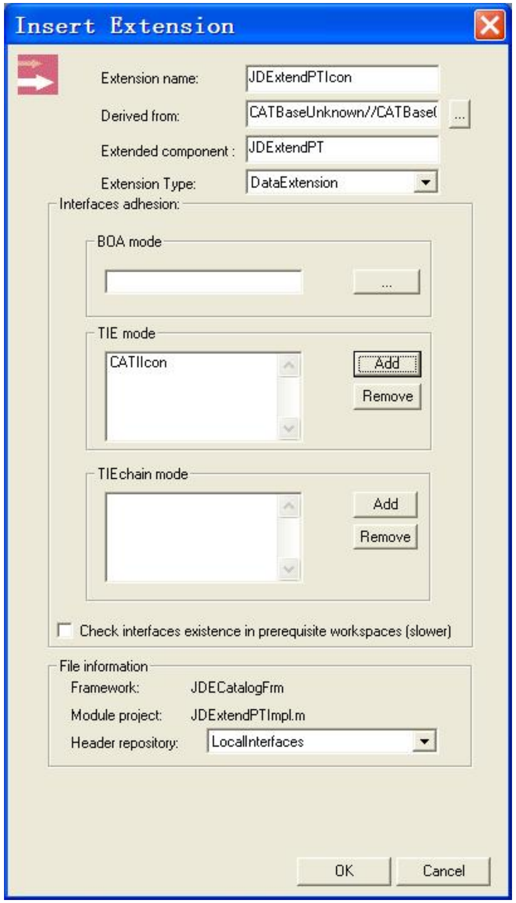
- 双击自定义特征/编辑自定义特征的处理

```
HRESULT rc=E_FAIL;
CATCommand *pCommand=NULL;
CATBaseUnknown_var EditedJDPT;
if(iPath) EditedJDPT=iPath->SearchObject(JDAIPoint::ClassName());
if( NULL_var != EditedJDPT ) {
  JDAIPoint *piPT=NULL;
  rc = EditedJDPT->QueryInterface(IID_JDAIPoint,(void **)&piPT);
  if( SUCCEEDED(rc) && NULL != piPT ) {
    pCommand = new JDExtendPTCmd(piPT);
    piPT->Release();   piPT = NULL;
  }
  EditedJDPT->Release();  EditedJDPT = NULL;
}
return pCommand;
```

建立Implementation

－ JDExtendPTIcon

- 建立自定义特征的编辑方法
  - 设定JDExtendPTImpl为启动项目
  - 建立一个Component Extension
  - Estension Name：JDExtendPTIcon
  - Extended component：JDExtendPT
  - Extension Type：DataExtension
  - TIE mode：CATIIcon



Insert Extension

Extension name: JDExtendPTIcon

Derived from: CATBaseUnknown//CATBase(

Extended component : JDExtendPT

Extension Type: DataExtension

Interfaces adhesion:

BOA mode

TIE mode

CATIIcon

TIEchain mode

□ Check interfaces existence in prerequisite workspaces (slower)

File information

Framework: JDECatalogFrm

Module project: JDExtendPTImpl.m

Header repository: LocalInterfaces

建立Implementation

— JDExtendPTIcon

- 编写CATIIcon接口中方法的具体实现：

GetIconName(CATUnicodeString &oName)：

- 设定自定义特征的图片名称

```
oName = CATUnicodeString(“JDACA_I_JDAPT”);
return S_OK;
```

SetIconName(const CATUnicodeString iName)：

- 不需在这个方法中做任何修改

```
return S_OK;
```

内容介绍

- 建立Catalog
- 建立Interface
- 建立Implementation
- 建立用户交互（UI）

建立用户交互

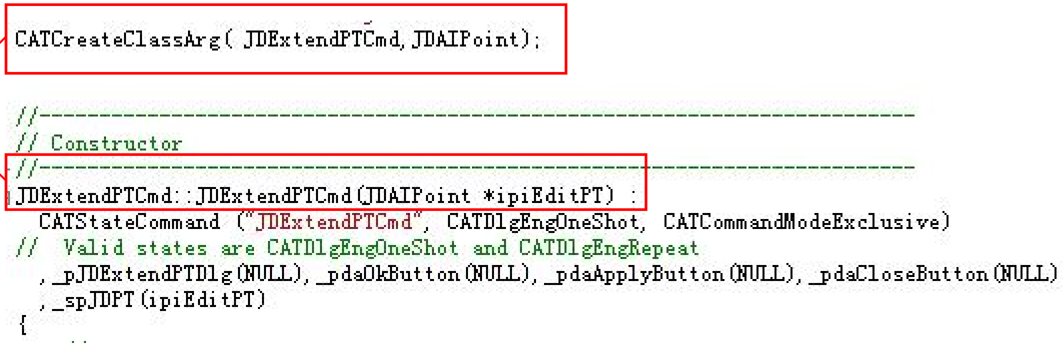
- 作用
  - 建立对话框（Dialog）
  - 建立命令（Command）
  - 建立工具条（Addin）
- 建立UI的Framework
  - Name：JDExtendUI
  - Type：Implementation
  - Function：Development
- 建立Module
  - 设定JDExtendUI为启动项目
  - Name：JDExtendUIM
  - Build mode：Shareed object

建立用户交互

- 建立对话框（Dialog）
  - 设定JDExtendUIM为启动项目
  - Type：Dialog box
  - Name：JDExtendPTDlg
  - 建立自定义特征输入属性的设定及获取方法，如：
GetValueX(),GetValueY(),GetValueZ()
SetValueX(),SetValueY(),SetValueZ()

- 建立命令（Command）
  - 设定JDExtendUIM为启动项目
  - Name：JDExtendPTCmd
  - Style：Statechart command
  - Mode：Exclusive

添加及修改这两处，以便确定是编辑还是创建自定义特征



```
CATCreateClassArg( JDExtendPTCmd,JDAIPoint);

//----------------------------------------------------------------------
// Constructor
//----------------------------------------------------------------------
JDExtendPTCmd::JDExtendPTCmd(JDAIPoint *ipiEditPT) :
  CATStateCommand ("JDExtendPTCmd", CATDlgEngOneShot, CATCommandModeExclusive)
//  Valid states are CATDlgEngOneShot and CATDlgEngRepeat
  ,_pJDExtendPTDlg(NULL),_pdaOkButton(NULL),_pdaApplyButton(NULL),_pdaCloseButton(NULL)
  ,_spJDPT(ipiEditPT)
{
    ..
```

建立用户交互

- 建立Addin
  - 设定JDExtendUIM为启动项目
  - ……

谢 谢！