**获取文档零件**

HRESULT hr;

//获取当前文档

CATDocument \*pDoc=NULL;

CATFrmEditor \*pEditor = CATFrmEditor::GetCurrentEditor();

//state command可直接获取GetEditor(); CATFrmEditor \*pEditor = GetEditor();

//通过CATDialogAgent中的GetEditor()获取Catia中已经打开的零件的窗口编辑的对应对象,

//每一个窗口都有独立的GetEditor();

CATILinkableObject \*pILinkableOnActive = NULL;

hr = pEditor->GetUIActiveObject().Search(IID\_CATILinkableObject,(void\*\*)&pILinkableOnActive);

//通过pEditor获取对应激活的节点,

//获取到的是一个CATPathElement的对象,通过CATPathElement对象获取到LinkableObject具体的对象

pDoc = pILinkableOnActive->GetDocument();

//获取当前激活节点所对应的文档

if(pDoc == NULL) return;

//获取文档容器

CATIContainerOfDocument\_var spDoc = (CATIContainerOfDocument\_var)pDoc;

CATIContainer \*piContainer = NULL;

if(FAILED(spDoc->GetSpecContainer(piContainer)))return;

//从文档容器中获取零件容器

CATIPrtContainer \*piPrtCont = NULL;

hr = piContainer->QueryInterface(IID\_CATIPrtContainer,(void\*\*)&piPrtCont);

if(FAILED(hr))return;

//获取零件

CATIPrtPart\_var spPart = piPrtCont->GetPart();

打开会话,创建Document

1--

// Check Argument

if ( argc < 2)

{

cout << " ERROR : Pass file name as argument" << endl;

return 1;

}

else cout << " Argument Checked OK." << endl << flush;

//char \*pFileName = "TSTScrewGeom";

char \*pFileName = argv[1];

// Create a Session

char \*sessionName = "TSTScrewSession";

CATSession \*pSession = NULL;

HRESULT hr = ::Create\_Session(sessionName, pSession);//::全局函数

if ((FAILED(hr)) || (NULL == pSession))

{

cout << "ERROR in creating session" << endl << flush;

return 2;

}

// Create a Document

CATDocument\* pDocument = NULL;

hr = CATDocumentServices::Open(pFileName, pDocument);

if ((FAILED(hr)) || (NULL == pDocument))

{

cout << "ERROR in creating New document" << endl << flush;

return 3;

}

// Remove Document

CATDocumentServices::Remove(\*pDocument);

// Delete Session

::Delete\_Session(sessionName);

2---

//Create a File

char\* pFileName = "SQMSaveFile";

//Create a Session

cout<<"== Create a Session =="<<endl;

char\* psessionName = "SQMScrewSession";

CATSession\* pSession = NULL;

HRESULT rc = Create\_Session(psessionName,pSession);

if (FAILED(rc)||NULL==pSession)

{

cout<<"create session error"<<endl;

return 1;

}

//Create a Document

cout<<"== Ceate a Document =="<<endl;

CATDocument\* pDocument = NULL;

rc = CATDocumentServices::New("Part",pDocument);

if(FAILED(rc)||NULL==pDocument)

{

cout<<"Create New Document error"<<endl;

return 2;

}

//Save Document

cout<<"== Save Document =="<<endl;

CATDocumentServices::SaveAs(\*pDocument,pFileName);

//Delete Session

cout<<"== Delete Session =="<<endl;

pSession->Delete\_Session(psessionName);

//构建path

CATIBuildPath \*piBuildPath=NULL;

rc=ispiSpecObject->QueryInterface(IID\_CATIBuildPath,(void\*\*) &piBuildPath );

CATPathElement contextPath=\_pEditor->GetUIActiveObject();

CATPathElement \*pPathElement = NULL ;

if( !!piBuildPath ) {

rc=piBuildPath->ExtractPathElement(&contextPath,&pPathElement);

piBuildPath->Release(); piBuildPath = NULL;

}