

第 2 讲 对话框控件和命令的调用机制

目录

1	概述	3
2	简介	3
2.1	对话框定义	3
2.2	工作空间 (Workspace)	3
3	对话框 (Dialog) 类	4
3.1	对话框 (Dialog) 类结构简介	4
3.2	控件简介	5
3.3	对话框 (Dialog) 组成	5
3.4	对话框 (Dialog) 控件的方法 (Method)	9
3.4.1	继承继承父类 (Father) 的方法 (Method)	9
3.4.2	控件函数自带方法 (Method)	9
3.5	界面排版	10
3.6	对话框 (Dialog) 事件简介	11
3.6.1	事件概述	11
3.6.2	回调函数	13
4	创建对话框流程	13
4.1	创建工作空间 (Workspace) 和框架 (Frame)	13
4.2	创建启动对话框命令 (Command)	13
4.3	创建响应对话框 (Dialog)	14
4.4	根据需求使用相应的控件	15
5	创建并编译命令 (Command) 和对话框 (Dialog)	17
5.1	模块 (Module) 检查	17
5.2	将命令 (Command) 与菜单 (Addin) 关联	18
5.3	将命令 (Command) 与对话框 (Dialog) 关联	18
5.4	空命令运行	19
5.5	对话框调用机制	21
5.6	代码添加方法	21
5.6.1	消息头指针声明	21
5.6.2	Combo 初始化	21
5.6.3	PushButtonGetName 响应函数	22
5.6.4	PushButtonGetGender 响应函数	23
5.6.5	PushButtonGetSubject 响应函数	24
5.6.6	PushButtonGetAge 响应函数	25
5.6.7	PushButtonGetHappyiness 响应函数	25
5.6.8	PushButtonAdd 响应函数	26
5.6.9	PushButtonDelete 响应函数	28
5.6.10	WindClose 响应函数	28
5.6.11	CATDIgFile 使用	28
5.6.12	MultiListInformation 的选择响应函数	30

5.7	完整实例运行	31
6	总结	31

1 概述

创建一个对话框，可以实现信息录入的功能，对话框如下图：

姓名	性别	喜欢学科	年
苏姐己	女	知识工程设计制造集成三维工艺...	24
康熙	男	知识工程三维工艺设计	21

学习控件 Editor、PushButton、Comb 等的用法

2 简介

2.1 对话框定义

这部分内容：百科全书->User interface ->WIntop Dialog (选项卡)里边内容。

里边介绍与 Dialog 有关的东西。

C++ 类的概念和用法。

2.2 工作空间 (Workspace)

Train2 中很大程度用到 Train1 所学知识。简单再解释一下文件结构。

如图：项目 BITAssemAddins.m 下的 BITAssemAddin.h 和 BITAssemAddin.cpp 文件用来创建菜单，和工具条。

项目 BITAssemFrmWork 下的文件一般不能动，是整体框架。

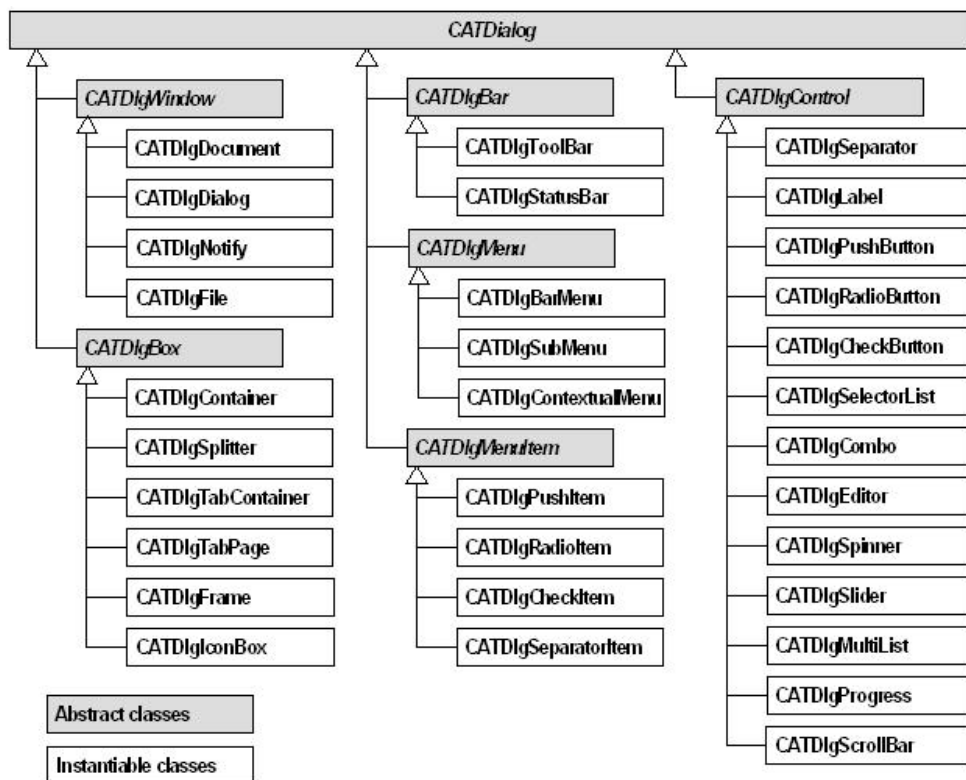
BITAssemCmds.m 下的 LocalInterfaces 文件夹里边存放的是.h 文件和.CATDlg 文件, 而 scr 里边存放.cpp 文件

在 CAA 的编程好中要大量使用字符串, CAA 自己定义了两种字符串 CATSting 和 CATUnicodeString。在 help view 中可以找的这些类的方法。字符串常用的操作有 BuildFromNum(数字转化为字符串); ConvertToChar(转化为 char*类型); ConvertToNum (转化为数字) GetLengthInChar(字符串的字符个数); Insert (字符串中插入字符串) ReplaceSubString (替换字符串) SearchSubString (查找字符串) Resize (调整字符串长度), Compare (比较字符串大小) SubString (获取一段字符串)

3 对话框 (Dialog) 类

3.1 对话框 (Dialog) 类结构简介

CATDialog 类的派生关系详细的在图中列出。



对话框框架主要定义了两种类: containers 和 components;


Containers 是用来容纳组件的容器, 包括 window、menu、dlgbox、barBox


Components 是构成对话框、窗口等的项目包括对话框控件类: CATDlgLabel (标签)、CATDlgPushButton(按钮)、CATDlgComb(下拉列表); 菜单项, CATDlgRadioItem (菜单选项)、CATDlgRadioItem (菜单分隔符)。


3.2 控件简介


打开 CATDlg 文件。看到如下所示控件集：





: frame 控件，同 C++ 中的 Frame 一样。Frame（框架）控件可以用来对其他控件进行分组，以便于用户识别。


: Tab Container，对话框中的选项卡，就是用它来实现。


: label，如同名字，就是一个标签。这个标签也可以是一个 Icon（图片）。


: Push button，按钮。


: Radio Button，一组 Radio button 只能选择一个。比如：男和女，只能选择一个。


: Check Button，与 Radio Button 相对，一组 Check Buton 可以多选，可以单选。

: Spinner，可以选择直接输入数据，亦可以按向上或者向下的箭头输入数据。

: Multilist，可以分多栏，一般用来显示数据。

: Editor，输入数据。

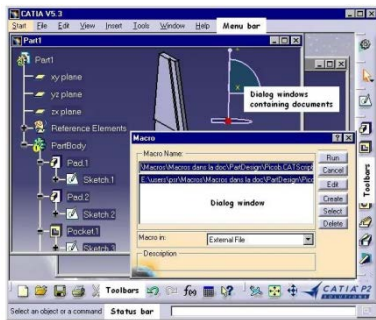
: 这是让编程人员更好操作编辑对话框，功能就是显示红色网格线。

separator 对话框分割，用来美化对话框，让对话框看着有条理。

 Combo 下拉列表。

3.3 对话框（Dialog）组成

Window:



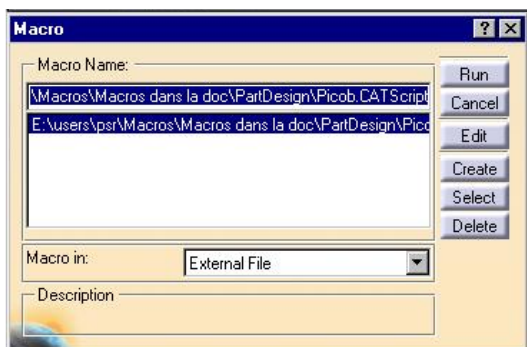
文件对话框类：CATDlgFile



消息对话框类 CATDlgNotify

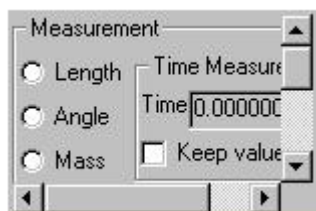


对话框类 CATDlgDialog

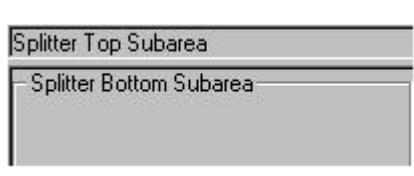


Box

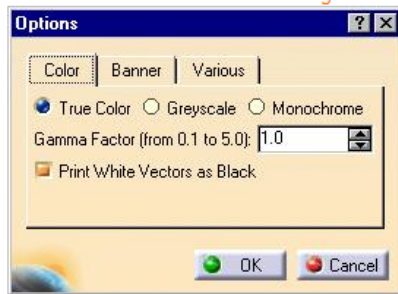
CATDlgContainer



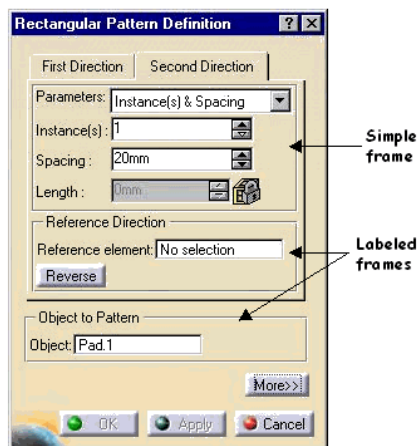
CATDlgSplitter



选项卡 CATDIgTabContainer

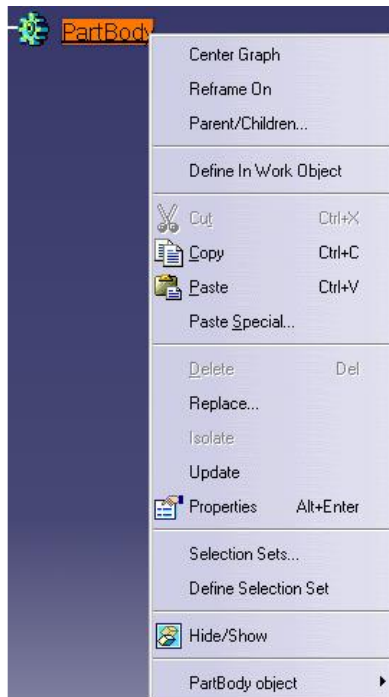


框 CATDIgFrame



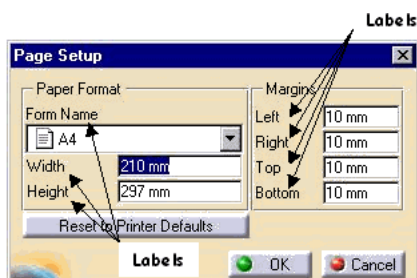
Menu

右键快捷菜单 CATDIgContextualMenu



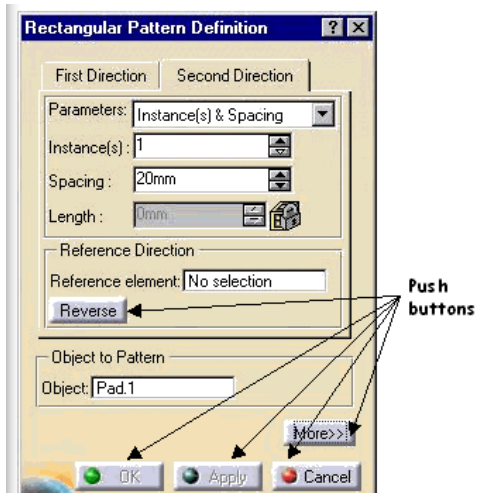
Indicators 替其他控件命名

标签 CATDlgLabel



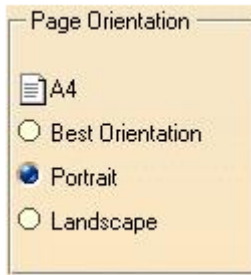
触发响应的控件

按钮 CATDlgPushButton

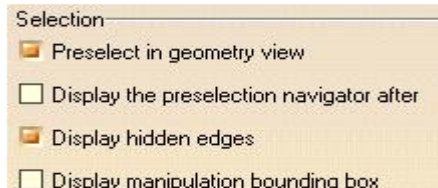


设置选项的控件

单项选择 CATDlgRadioButton



多项选择 CATDlgCheckBox



3.4 对话框（Dialog）控件的方法（Method）

3.4.1 继承继承父类（Father）的方法（Method）

由于 CATDlgControl 与具体的控件之间的为继承关系，所以 CATDlgControl 的方法具体的控件也可以调用。比如：设置对话框控件的图标 CATDlgControl::SetIconBitmap(const unsigned char* iBitmap, intiWidth, intiHeight) ；

另外 CATDlgControl 继承了 CATDialog 的一些方法，这些方法在具体的控件也可以调用。比如：设置对话框控件是否可用的方法 CATDialog::SetSensitivity(CATULong)；设置对话框控件是否隐藏的方法 CATDialog::SetVisibility(CATULongiState)；设置显示标题的方法 SetTitle(CATUnicodeString&);

3.4.2 控件函数自带方法（Method）

获取 CheckButton 的状态 CATDlgCheckBox::GetState();

设置 CheckButton 的状态 CATDlgCheckBox::SetState(CATULong,int) ;

获取 RadioButton 的状态 CATDlgRadioButton::GetState();

设置 RadioButton 的状态 CATDlgRadioButton::SetState(CATULong,int) ;

获取 Combo 的某行 CATDlgComb::GetLine() 有很多重载函数

设置 Combo 的某行 CATDlgComb::SetLine()有很多重载函数

设置 Combo 的选择的行 CATDlgComb::SetSelect () 有很多重载函数

获得 Combo 的选择的行 CATDlgComb::GetSelect() 有很多重载函数

清除 Combo 行中的内容 CATDlgComb::ClearLine () 有很多重载函数

获取 Editor 的某行 CATDlgEditor::GetLine() 或者 GetText();GetValue ();

设置 Editor 的某行 CATDlgEditor::SetLine(), SetText();SetValue();

设置 MultiList 的某行 CATDlgMultiList::SetColumnItem (XXX);

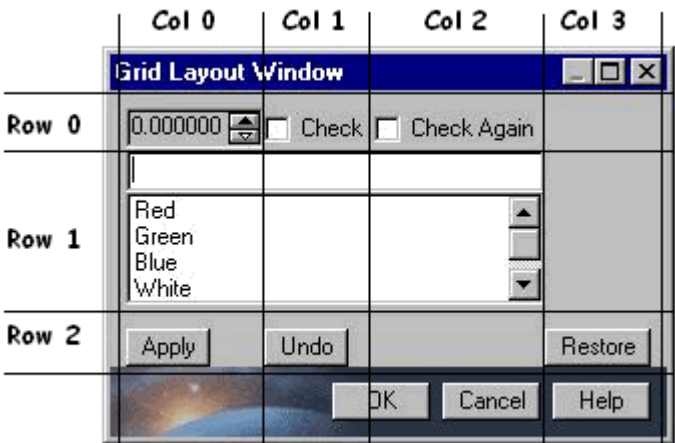
常用的方法还有设置选择, 判断某行是否选择, 读取某行的值, 设置 MultiList 可编辑模式, 情况 MultiList 中的内容, 设置可以看到的行数等。

3.5 界面排版

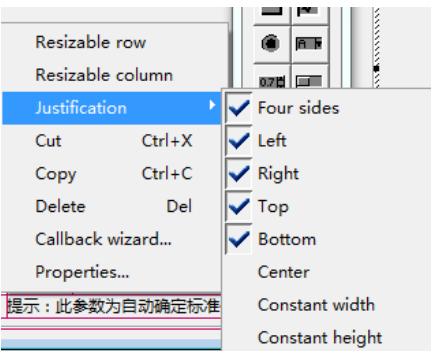
完成文件->add CAA V5 Item->CATIA resource->Dialog 对话框创建后自动创建一个继承自 CATDlgDialog 的对话框类。并且会自动启动 CATCAADialogbuilder 进程, 这是 CAA 支持对话框交互设计的一个插件, 在 VS 中双击打开后缀名为.CATDlg 的文件后也会启动这个进程, 有了这个进程用户可以交互设计对话框。

CATCAADialogbuilder 采用的是网格布局设计, 一个控件初始情况下均占用一个格子。

设计对话框时只需要将控件拖入到对话框中即可。



对某个控件右键点击, 在 justification 中可以选择停靠的模式。



双击控件或者在控件的右键菜单的 Properties 中可以修改控件的属性。所有的控件都可以修改 Name; Label、Pushbutton、Radiobutton、CheckButton 等控件可以修改 Title (界面显示)。

MultiList 可以设置列数目，和列的标题，以及是否多项选择等属性；

Editor 可以设置文本的长度、输入是 string/double/float/int/密码等、物理意义、度量单位、每个控件的属性都不同。

有些对话框的属性在对话框布局设计时不能给定，需要用程序设置，比如是否显示，是否可以编辑等，MultiList 的可视行数等。

在对话框空白的区域双击或者右键的方式进入 Properties 页面可以设置对话框的属性，包括 Name、Title、是否模态对话框、OK\Apply\Cancel\Help\Close 键的情况等。

模态对话框（Modal window），是指在用户想要对对话框以外的应用程序进行操作时，必须首先对该对话框进行响应。如单击【确定】或【取消】按钮等将该对话框关闭。

一般与图形进行交互的对话框必须选择非模态对话框。

设计完成后保存.CATDlg 文件，若.h 文件和.cpp 文件都是打开的状态，则会在两个文件中自动生成对话框布局设计代码。一般情况下不要在自动生成的代码中写入自己的程序，否则在重新保存.CATDlg 文件的时候自动添加代码重新生成，用户添加的代码会被删除掉。

保存.CATDlg 文件时会在 framework 的 CNext\resources\msgcatalog 中自动产生两个名称为对话框名的资源文件并保存，与创建 Addin 时一样，其中 CATNls 文件存储对话框所有的汉字标题，CATRsc 文件存储用到的图片信息等。如图：

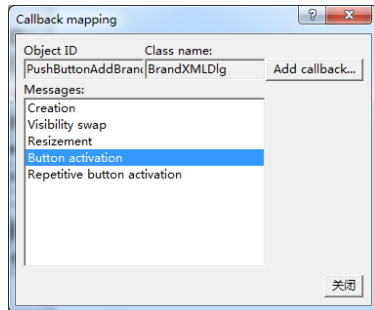
```
// DO NOT EDIT :: THE CAA2 WIZARD WILL REPLACE ALL THE CODE HERE
Title = "标准件标准发布"
Frame036.Title = "外部参数";
Frame036.Label038.Title = "参数名称";
Frame036.Label050.Title = "搜索规则";
Frame036.Label057.Title = "提示：此参数为自动确定标准件牌号的依据";
Frame036.Label088.Title = "参数名称";
Frame036.Label090.Title = "搜索规则";
Frame036.AddExportParams.Title = "添加";
Frame036.AddExportParam2.Title = "添加";
Frame036.Label109.Title = "Auto(系统自动确定的)";
Frame036.Label110.Title = "Interactive(交互式)";
Frame093.Title = "参数";
Frame093.InternalParamName.Title = "参数名称";
Frame093.AddInternalParams.Title = "添加";
Frame098.AddVoxel.Title = "体素建模";
Frame098.PushButton077.Title = "录入牌号";
Frame098.PushButton078.Title = "确定";
Frame098.PushButton080.Title = "取消";
Frame002.Title = "标准件基本信息";
Frame002.Label005.Title = "标准件类型";
Frame002.Label012.Title = "标准名代号";
Frame002.PushButton082.Title = "创建";
Frame002.ClearSuffix.Title = "清空";
Frame002.Label107.Title = "标准名称";
Frame104.Title = "标准件库";
Frame104.StandardSearchButton.Title = "查询";
Frame104.PushLibraryManagementButton.Title = "标准件库管理";
Frame104.Label106.Title = "标准名代号";
// END WIZARD REPLACEMENT ZONE
```

3.6 对话框（Dialog）事件简介

3.6.1 事件概述

事件就是对窗口上各种控件的操作，这儿可以是用户操作对话框控件，也可以是程序代替人进行操作。

在设计对话框布局的时候，选中一个对话框控件，并点击右键弹出快捷菜单，其中 Callback Wizard 就是编译器自动添加事件响应函数的选项，点击 Callback Wizard 进入 callback mapping 对话框：



不同的控件事件不同，上图是一个 PushButton 的事件，可以看到他的事件都有 Creation（创建）、Visibility swap（显示隐藏切换）、Resizement（重设大小）、Button Activation（按钮按下激活）等，其中按钮按下激活事件是最最常用的事件。

其他控件的常用事件还有：

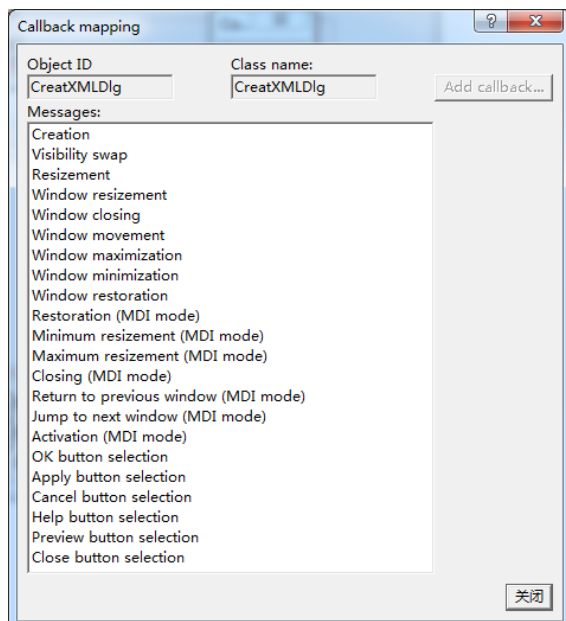
Editor 的 Edit Field Modification（输入发生变化）事件；

Combo、MultiList 的 Selection in list（在列表中选择）事件；

Combo、Editor、Spinner 的 Verify input（输入确认）事件；

CheckBox、RadioButton 的 Button modification（选择状态改变）事件。

在对话框的空白区域右键单击出现的 Callback wizard 中可以添加整个对话框的事件响应函数，这些事件包括 Window Closing（对话关闭，就是对话框右上角的 X）、Window Movement（拖动对话框）Ok Button selection（单击 Ok\确定按钮）Apply button selection（单击应用按钮）等。如下图所示：



3.6.2 回调函数

选择一个事件，单击 **add callback**,再次点击 **OK**，完成对话框事件响应函数的添加。

此时会在三个地方修改对话框代码，对话框.h.cpp 两个文件都有修改。

1.在 h 文件中会自动添加代码：

```
virtualvoid OnPushLibraryManagementButtonPushBActivateNotification (CATCommand*, CATNotification* , CATCommandClientData data);
```

这句函数声明了 **PushLibraryManagementButton** 的按钮响应函数。

2.在 cpp 文件的 **build()**函数中自动添加按钮响应函数的声明代码：

```
AddAnalyseNotificationCB (_PushLibraryManagementButton,
_PushLibraryManagementButton->GetPushBActivateNotification(),
(CATCommandMethod)&CreatXMLDlg::OnPushLibraryManagementButtonPushBActivateNotification, NULL);
```

3.在 cpp 中还会添加事件响应的具体回调函数：

```
void
CreatXMLDlg::OnPushLibraryManagementButtonPushBActivateNotification(CATCommand*
cmd, CATNotification* evt, CATCommandClientData data)
{
    // Add your code here
}
```

事件回调函数一般需要自己写。

值得注意的是在对话框布局中删除了某个控件，以前给这个控件添加的事件响应回调函数不会自动删除，需要手动删除。删除时要注意上述三部分自动添加的代码都要删除，否则会出错。

4 创建对话框流程

4.1 创建工作空间（Workspace）和框架（Frame）

根据第一章《CAA 简介和框架》，创建工作空间和框架，并添加相应的模块（module），用来存放命令（Command）和对话框（Dialog），此处不再赘述。

4.2 创建启动对话框命令（Command）

单击：文件->Add CAAV5 Item->CATIA resource->Command

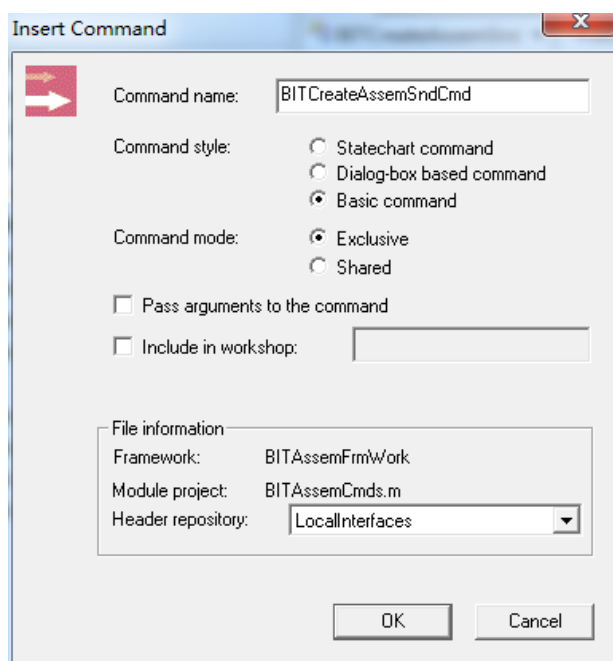
在弹出 **Insert Command** 对话框中输入和选择：

Command name: BITCreateAssemSndCmd（注意：此处名字必须和 BITAssemAddin.cpp 中 new BITAssemHeader()中的倒数第二个参数相同。BITAssemAdding.cpp 是根据这个参数运行 BITCreateAssemSndCmd 的）

Command style 选择 Basic Command

Command Mode 选择 Exclusive（默认值）

其他不用选择。如下图：



单击 OK 后生成两个文件：BITCreateAssemSndCmd.h 和 BITCreateAssemSndCmd.cpp 分别在 LocalInterfaces 和 src 文件夹下。

先不写入代码，将所有的文件添加全了再写代码

4.3 创建响应对话框（Dialog）

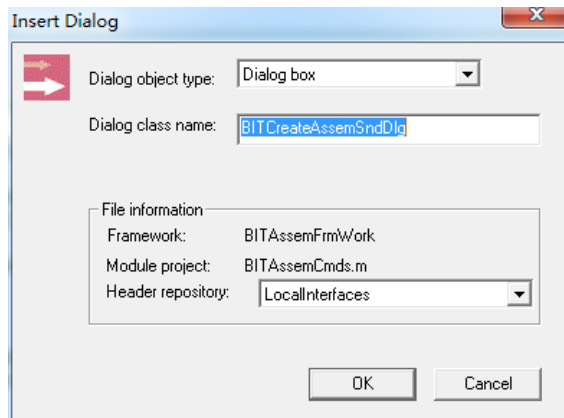
单击：文件->Add CAAV5 Item->CATIA resource->Dialog

在弹出 Insert Dialog 对话框中输入和选择：

Dialog object type: Dialog box（默认值）

Dialog Class name: BITCreateAssemSndDlg

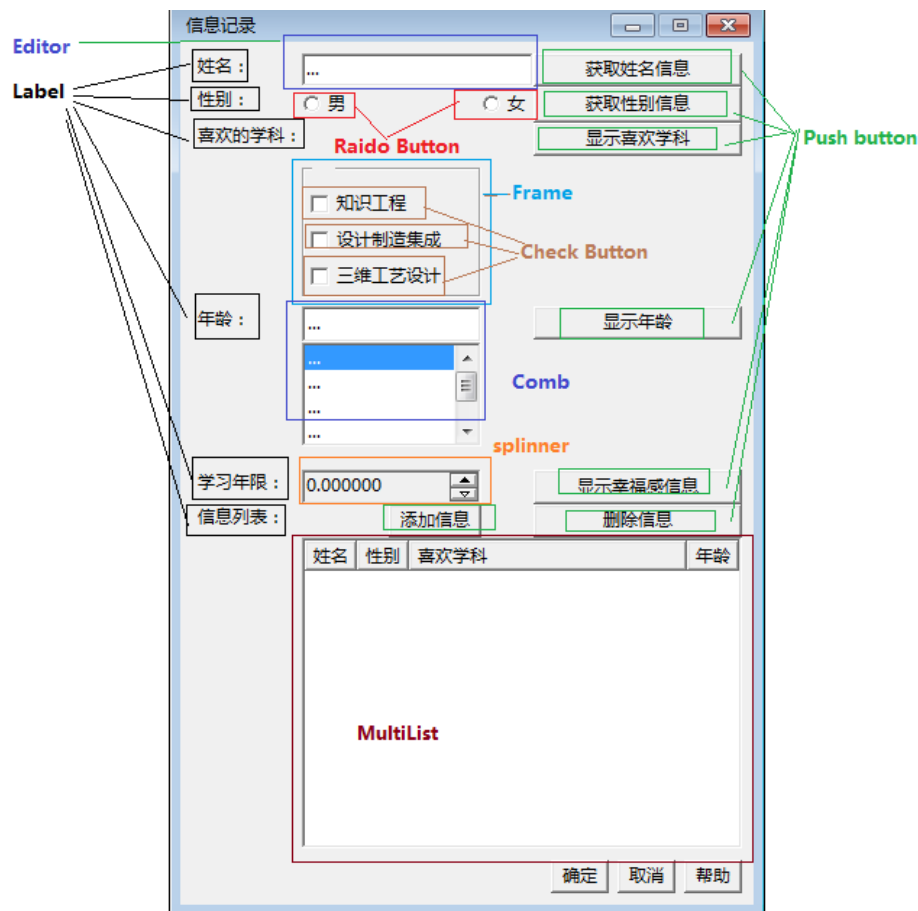
其他不用管。如下图：



单击 Ok 后，生成三个文件：在 LocalInterface 中的 BITCreateAssemSndDlg.h 和 BITCreateAssemSndDlg.CATDlg 以及在 src 中的 BITCreateAssemSndDlg.cpp

4.4 根据需求使用相应的控件

在对话框中拖入上述控件，并按照下图布置：



做对话框的时候要注意：不要一次拖入多个控件，应该拖入一个控件修改控件的属性然后再拖入下一个控件。

修改所有控件的 name 和 Title 属性：

在控件上双击或者右击在弹出的对话框中选择 property。

一般而言，Name 中修改变量名称，Title 修改显示在对话框的名称。

以下的 Name 前加了横线，在属性里边修改的时候去掉。

Title 可以随意但是 name 一定要和后边程序里的内容一致。

例如：“姓名：” label 控件的修改：

双击打开 label 控件属性对话框。

从下图可以看到 Title 为“姓名：”（绿色列）；Name 为“label001”（中间列，去掉开头的下划线）

控件类型	name	//Title
CATDlgLabel*	_Label001;	//姓名：
CATDlgLabel*	_Label002;	//性别：
CATDlgLabel*	_Label003;	//喜欢学科：
CATDlgEditor*	_EditorName;	
CATDlgPushButton*	_PushButtonGetName;	//获取姓名信息
CATDlgPushButton*	_PushButtonGetGender;	//获取性别信息
CATDlgRadioButton*	_RadioButtonMale;	//男
CATDlgRadioButton*	_RadioButtonFemale;	//女
CATDlgPushButton*	_PushButtonGetSubject;	//显示喜欢学科
CATDlgFrame*	_Frame010;	//“ ”
CATDlgCheckButton*	_CheckButtonKnowledge;	//知识工程
CATDlgCheckButton*	_CheckButtonDesign;	//设计制造集成
CATDlgCheckButton*	_CheckButton3D;	//三维工艺设计
CATDlgLabel*	_Label014;	//年龄：
CATDlgCombo*	_ComboAge;	
CATDlgPushButton*	_PushButtonGetAge;	//显示年龄
CATDlgLabel*	_Label017;	//学习年限：
CATDlgSpinner*	_SpinnerYear;	
CATDlgPushButton*	_PushButtonGetHappyness;	//显示幸福感信息
CATDlgLabel*	_Label020;	//信息列表：
CATDlgPushButton*	_PushButtonDelete;	//删除信息
CATDlgPushButton*	_PushButtonAdd;	//添加信息
CATDlgMultiList*	_MultiListInformation;	

MultiList 控件属性修改：

修改 name 为 MultiListInformation 之后点击应用。

单击 Attribute 选项卡修改列数。应用

单击 resource 选项卡修改字段 Title

点击应用后看对话框是否是你想要的效果，如果不是，再次修改

（注意：这个编辑对话框的内部服务很脆弱，经常会出现一下三种问题：

控件不能拖入对话框

拖入后不能打开属性从而修改控件 name 和 title

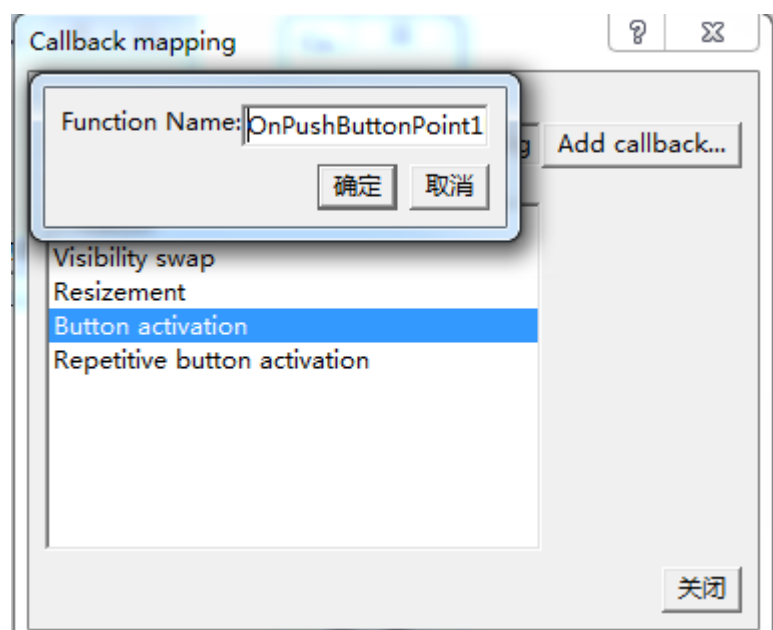
将完成的 CATDlg 文件保存后再次打开，出现在屏幕上的是编码而非对话框。

出现以上三个问题之后，重启电脑，然后重做吧。（目前只能这样）

最后，保存并关闭 CATDlg 文件）

给 PushButton 添加回调函数：做完上述对话框布局后给每个 PushButton 添加 callback

右击 Push button 控件选择 Callback wizard。弹出 CallBack mapping 对话框：



选择 Button activation，单击 AddCallBack，单击确定。

给对话框添加回调函数：

右击对话框的空白处，选择 Callback wizard。

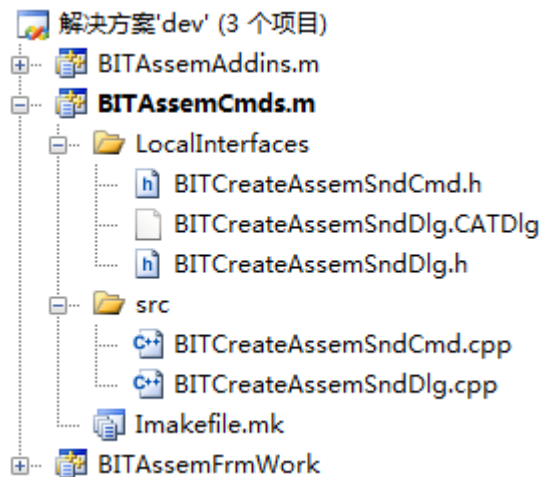
在弹出的对话框中给 WindClose、CLOSE、OK、APPLY 按钮添加回调函数，这个必须添加，否则在 BITCreateAssemSndDlg.cpp 文件中找不到添加相应函数的地方。

所有的控件修改完毕后保存，关闭.CATDlg。

5 创建并编译命令（Command）和对话框（Dialog）

5.1 模块（Module）检查

做到这一步时首先检查 BITAssemCmds.m 中是否有如下图所示的文件：



这五个文件一定要有。

5.2 将命令（Command）与菜单（Addin）关联

在 BITAssemAddin.cpp 中 new BITAssemHeader()函数的第三个变量必须和命令名一致。
具体见 BITAssemAddin.cpp 的代码注释。

这样就将 Cmd 和 Adding 关联了。

5.3 将命令（Command）与对话框（Dialog）关联

打开 BITCreateAssemSndCmd.cpp 文件，找到函数：

```
CATStatusChangeRC BITCreateAssemSndCmd::Activate( CATCommand * iFromClient,
CATNotification * iEvtDat)
{
    return (CATStatusChangeRCCompleted);
}
```

在大括号中的 return 前添加如下代码：

```
//命令响应菜单的单击操作，执行所需的命令
BITCreateAssemSndDlg* pDlg = new BITCreateAssemSndDlg();
pDlg->Build();
pDlg->SetVisibility(CATDlgShow);
RequestDelayedDestruction();
```

添加头文件

```
#include"BITCreateAssemSndDlg.h"
```

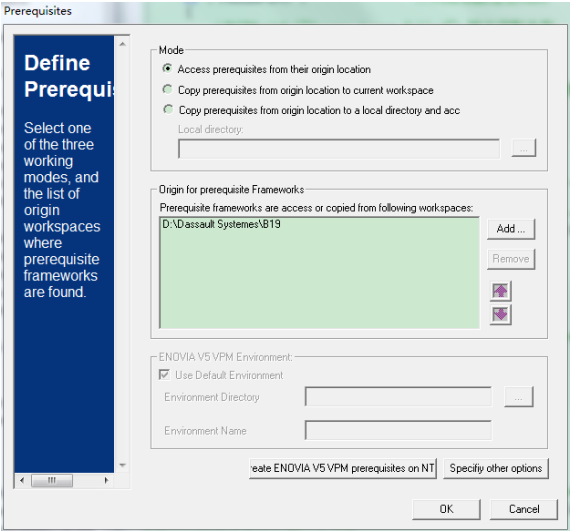
这些在 Train1 中学习过，用来新建并打开对话框的。

具体看 BITCreateAssemSndCmd.cpp 代码注释

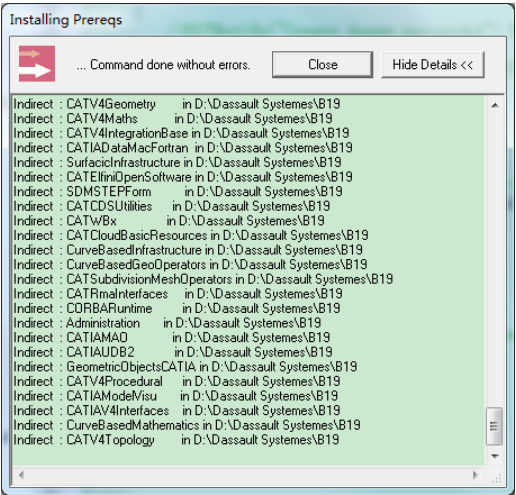
5.4 空命令运行

然后进行以下 A.B.C...步骤:

A. 单击 CAAV5 Workspace->Local prerequisite workspace 弹出如下对话框:

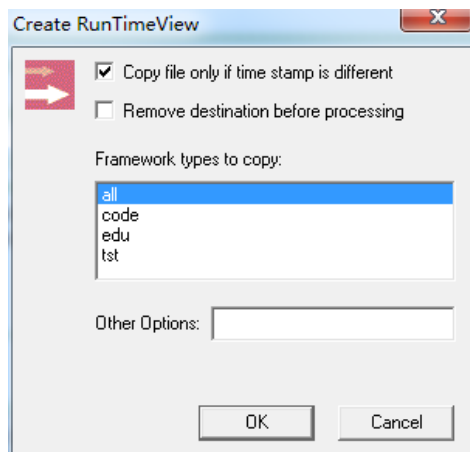


单击 OK 后:



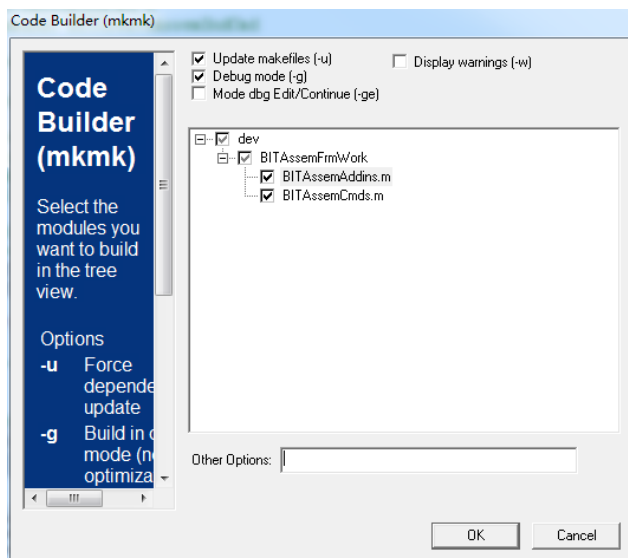
单击 Close。

B. 单击 CAAV5 Workspace->Create/Update Runtime view 弹出如下对话框:



单击 OK 后运行。

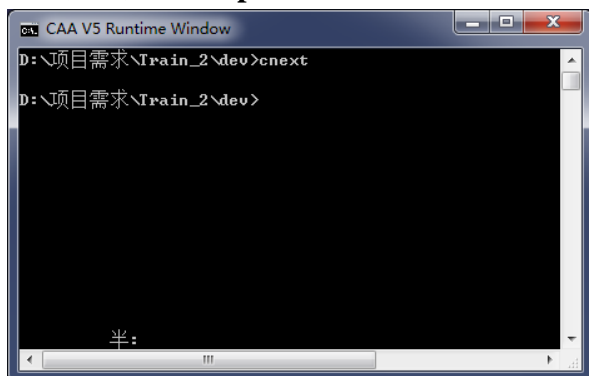
单击生成->mkmk 弹出如下对话框：



单击 OK 后开始运行。在结果输出栏里边没有 ERROR 的情况下才可进行下一步。

（注意：单击 OK 前要确保你的 CATIA V5 R19 关闭）

C.单击窗口->Open Runtime Window



在弹出对话框中输入：cnext 并按回车键。

D.单击装配 a->信息录入

弹出你所设计的信息记录对话框。只是由于没有添加各个控件的命令相应函数，所以没有任何功能。

5.5 对话框调用机制

用户在点击 `addin` 添加的菜单命令或工具条中的工具时首先激活了 `BITCreateAssemSndCmd` 命令的 `Active` 相关的函数 `BITCreateAssemSndCmd::Activate(xxxxxxx)`；进入了对话框的构造函数 `BITCreateAssemSndDlg()`，声明对话框控件；

进入了对话框的 `Build()` 函数，完成了对话框控件的创建控件布局 and 实现回调函数的声明；

然后显示对话框。`pDlg->SetVisibility(CATDlgShow)`；显示后，对话框等待用户的事件（事件包括对话框拖动、缩放、按钮的按下，键盘的输入，`Editor` 内容的更改、`Combo` 的选择等）；

当有事件发生时处理事件列表，开始响应事件环节，从对应的事件声明中调用事件的回调函数。处理完一个事件的全部内容后才会开始下一个事件的处理，两个事件不会同时处理。

完成列表中的所有事件后回到对话框再次等待事件。

若某次事件执行过程中出现对话框关闭函数（`this->SetVisibility(CATDlgHide)`）；则关闭对话框。

开始析构函数调用 `~BITCreateAssemSndDlg()`；带~的为析构函数。所有的内存会在析构函数中释放。

这儿需要注意构造函数 `BITCreateAssemSndDlg()`、对话框的 `Build()`、析构函数 `~BITCreateAssemSndDlg()` 是自动产生的，一般不用编辑。而命令激活函数 `BITCreateAssemSndCmd::Activate (xxxxxxx)`、事件的回调函数是需要添加代码才能完成具体功能。

另外，事件列表的概念需要注意。比如 `A` 事件的回调函数中出现了另外一个事件 `B`，此时继续执行 `A` 回调函数到最后再执行 `B` 的回调函数。

5.6 代码添加方法

5.6.1 消息头指针声明

打开 `BITCreateAssemSndDlg.h`

在 `//END CAA2 WIZARD WIDGET DECLARATION SECTION` 后添加代码：

```
CATDlgNotify * _OpenNotify;           //对话框CATDlgNotify的声明
```

5.6.2 Combo 初始化

打开 `BITCreateAssemSndDlg.cpp`

在 `//END CAA2 WIZARD WIDGET CONSTRUCTION SECTION` 之后

```
//CAA2 WIZARD CALLBACK DECLARATION SECTION之前添加如下代码:
////////////////////////////////////对Combo进行赋初值(初始化),
CATUnicodeStringistring[30];
int i;
for(i=0;i<30;i++)
{
istring[i].BuildFromNum(i+20,"%d");           //将数字转换为字符串函数BuildFromNum
_ComboAge -> SetLine(istring[i],i);           //赋值函数SetLine
}
_ComboAge -> SetSelect(1);                     //Combo进行初始选择, 刚开始选择第二行
////////////////////////////////////
```

5.6.3 PushButtonGetName 响应函数

找到 PushButtonGetName 的 Callback 函数:

```
voidBITCreateAssemSndDlg::OnPushButtonGetNamePushBActivateNotification(CATCommand*
cmd, CATNotification* evt, CATCommandClientData data)
{
```

这个函数是用来响应单击 PushButtonGetName 事件。在大括号中写入:

```
    CATUnicodeString iText;           //定义一个字符串
    iText =_EditorName->GetText(); //从EditorName中获得字符串并赋给iText
```

(注意: 此处并没有写 if{} else{} 语句里的内容, 而源代码里边有)

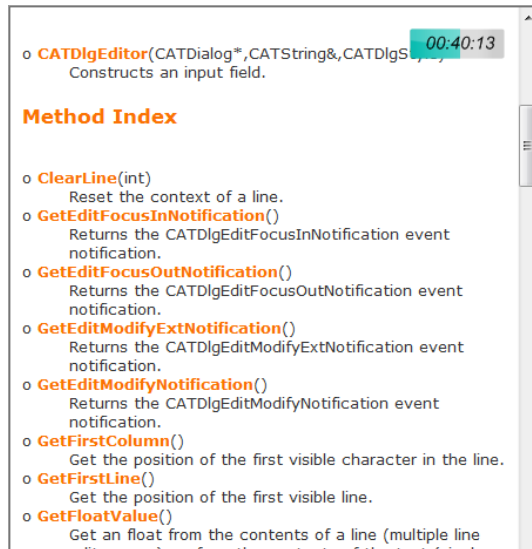
可以查 Help 文档看 GetText 的详细介绍。

从相应的 BITCreateAssemSndDlg.h 文件的声明中可以看到:

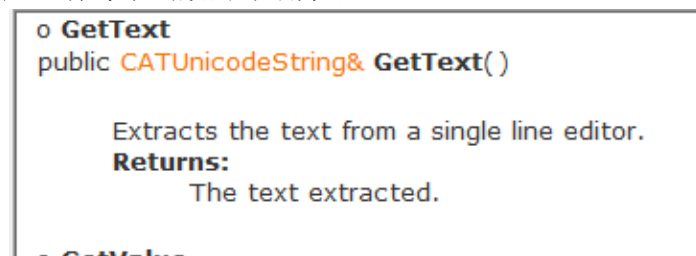
```
CATDlgEditor*      _EditorName;
```

从这句代码中得知 _EditorName 是一个指向 CATDlgEditor 的指针。那么所有要对 _Editor 操作的函数都应该在 CATDlgEditor 中找。所以按照 6.1 中 C.所述方法搜索 CATDlgEditor。

搜索结果如下:



后边有每个函数的详细介绍：



从中可以看到该函数的返回值等信息。

同理，上述代码中的函数：BuildFromNum、SetLine、SelectLine 都可以找到。

以后用到的函数我只是简单的介绍，不会这么详细的叙述查找方法。

5.6.4 PushButtonGetGender 响应函数

找到void

```
BITCreateAssemSndDlg::OnPushButtonGetGenderPushBActivateNotification(CATCommand* cmd,
CATNotification* evt, CATCommandClientData data)
{
```

在大括号中添加代码：

```
if (_RadioButtonMale->GetState() != CATDlgCheck &&
_RadioButtonFemale->GetState() != CATDlgCheck) //两个选项都没有选中
{
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
    _OpenNotify->SetText("请您选择性别!");
    _OpenNotify->SetVisibility(CATDlgShow);
}

elseif (_RadioButtonMale->GetState() == CATDlgCheck) //选中男
{
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
    _OpenNotify->SetText("您选择的性别是：\n男");
}
```

```

        _OpenNotify->SetVisibility(CATDlgShow);
    }

    else
    {
        _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
        _OpenNotify->SetText("您选择的性别是: \n女");
        _OpenNotify->SetVisibility(CATDlgShow);
    }

```

GetState 是获得 RadioButton 的状态的函数，所以用来判断是否选中。

5.6.5 PushButtonGetSubject 响应函数

找到void

```

BITCreateAssemSndDlg::OnPushButtonGetSubjectPushBActivateNotification(CATCommand* cmd,
CATNotification* evt, CATCommandClientData data)
{

```

在大括号中添加：

```

    //用if else判断是否有输入. 没有选择学科. .有喜欢的学科。
    if (_CheckButtonKnowledge->GetState() !=CATDlgCheck &&
    _CheckButtonDesign->GetState() !=CATDlgCheck && _CheckButton3D->GetState() !=CATDlgCheck )
    {
        _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
        _OpenNotify->SetText("请您选择至少一门喜欢的学科！");
        _OpenNotify->SetVisibility(CATDlgShow);
    }

    else{

CATUnicodeString iText[3];    //定义一个三维字符串
    int i=0;

    if(_CheckButtonKnowledge->GetState() ==CATDlgCheck)
    {
        iText[i++] = "知识工程";
    }
    if (_CheckButtonDesign->GetState() ==CATDlgCheck)
    {
        iText[i++] = "设计制造集成";
    }
    if(_CheckButton3D ->GetState() ==CATDlgCheck)
    {
        iText[i++] = "三维工艺设计";
    }

```



```

    }
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
    _OpenNotify->SetText("您喜欢的学科是：
\n"+iText[0]+" \n"+iText[1]+" \n"+iText[2]+" \n"); //此次用到了字符串操作的“+”号
和\n换行符
    _OpenNotify->SetVisibility(CATDlgShow);
}

```

5.6.6 PushButtonGetAge 响应函数

找到void

```

BITCreateAssemSndDlg::OnPushButtonGetAgePushBActivateNotification(CATCommand* cmd,
CATNotification* evt, CATCommandClientData data)

```

```

{

```

在大括号中添加：

```

//获得选择的行号，并将选择的文本放在iText中。

```

注：对于局部变量iText，在这个函数执行完后释放内存，不会对后边的局部变量iText产生影响。

```

CATUnicodeString iText;

```

```

int LineSelected = _ComboAge->GetSelect() ; //获得选中的行号；注意第一行的行
号为0

```

```

_ComboAge->GetLine(iText, LineSelected); //将LineSelected对应的行中字符
串提出，放入iText

```

```

_OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);

```

```

_OpenNotify->SetText("您的年龄是： \n"+iText);

```

```

_OpenNotify->SetVisibility(CATDlgShow);

```

5.6.7 PushButtonGetHappyness 响应函数

找到void

```

BITCreateAssemSndDlg::OnPushButtonGetHappynessPushBActivateNotification(CATCommand* cmd,
CATNotification* evt, CATCommandClientData data)

```

```

{

```

在大括号中添加：

```

//获得SpinnerYear中的值，并根据相
应的值输出幸福感。用if 语句判断四种情况，并显示相应的信息。

```

```

int i=_SpinnerYear->GetCurrentValue(); //获得SpinnerYear中的值

```

```

if(i==0)

```

```

{
_OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);

```

```

_OpenNotify->SetText("请选择学习年限！");

```

```

_OpenNotify->SetVisibility(CATDlgShow);

```

```

}

```

```

elseif(i<3)

```

```

{

```

```

_OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);

```

```

        _OpenNotify->SetText("3年以下非常幸福!");
        _OpenNotify->SetVisibility(CATDlgShow);
    }
    elseif(i>6)
    {
        _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
        _OpenNotify->SetText("6年以上非常悲剧!");
        _OpenNotify->SetVisibility(CATDlgShow);
    }
    else
    {
        _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
        _OpenNotify->SetText("3到年之间马马虎虎!");
        _OpenNotify->SetVisibility(CATDlgShow);
    }
}

```

5.6.8 PushButtonAdd 响应函数

找到void

BITCreateAssemSndDlg::OnPushButtonAddPushBActivateNotification(CATCommand* cmd, CATNotification* evt, CATCommandClientData data)

{

在大括号中添加:

```

        //////////////////////////////////////////////////设置两个开关变量, CHECK来记录信息是否完整, CHECK2用来记录是否重名。当然可以将这两个变量定义成bool型。
        int CHECK=0, CHECK2=0;
        CATUnicodeString oCellContents[25];
        //////////////////////////////////////////////////判断是否所有的信息都填写
        if(_EditorName->GetText()=="")
            CHECK=1;
        if(_RadioButtonMale->GetState() !=CATDlgCheck &&
        _RadioButtonFemale->GetState() !=CATDlgCheck)
            CHECK=1;
        if(_CheckBoxKnowledge->GetState() !=CATDlgCheck &&
        _CheckBoxDesign->GetState() !=CATDlgCheck && _CheckBox3D->GetState() !=CATDlgCheck )
            CHECK=1;
        if(_SpinnerYear->GetCurrentValue()==0)
            CHECK=1;
        //////////////////////////////////////////////////判断是否有相同的名字已经添加在MultList中
        for(int i=0; i<_MultiListInformation->GetLineCount(); i++)
        {
            _MultiListInformation->GetColumnItem(0, oCellContents[i], i);
            if(oCellContents[i]==_EditorName->GetText())
                CHECK2=1;
        }
    }

```

////////////////////////////////////if else if else 又来了, 1. 判断信息是否完整。
2. 判断是否有重名。3. 将信息添加在MultiList中

```
if (CHECK==1)
{
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
    _OpenNotify->SetText("请您完善信息");
    _OpenNotify->SetVisibility(CATDlgShow);
}
elseif (CHECK2==1)
{
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);
    _OpenNotify->SetText("已存在同名, 请重新输入");
    _OpenNotify->SetVisibility(CATDlgShow);
}
else
{
    ///////////////////////////////////将收集到要添加的信息放在ColumnTitles[4]中,
    CATUnicodeString ColumnTitles[4];
    ColumnTitles[0]=_EditorName->GetText();           //收集姓名信息
    if (_RadioButtonMale->GetState()==CATDlgCheck)    //收集性别信息
        ColumnTitles[1]="男";
    elseif (_RadioButtonFemale->GetState()==CATDlgCheck)
        ColumnTitles[1]="女";
    CATUnicodeString iText;           //收集学科信息。注意重载用算符+=。
    if (_CheckButtonKnowledge->GetState()==CATDlgCheck)
    {
        iText = "知识工程";
    }
    if (_CheckButtonDesign->GetState()==CATDlgCheck)
    {
        iText += "设计制造集成";
    }
    if (_CheckButton3D ->GetState()==CATDlgCheck)
    {
        iText += "三维工艺设计";
    }
    ColumnTitles[2]=iText;
    CATUnicodeString myText;           //收集年龄信息
    int LineSelected = _ComboAge->GetSelect() ;
    _ComboAge->GetLine(myText, LineSelected);
    ColumnTitles[3]=myText;
    ///////////////////////////////////在MultiList中添加信息用到的函数是
    SetColumnItems, 可能有更好用的函数!
```

```

        _MultiListInformation->SetColumnItems(0, ColumnTitles, 1, -1, CATDlgDataAdd);
        _MultiListInformation
->SetColumnItems(1, &ColumnTitles[1], 1, -1, CATDlgDataAdd);
        _MultiListInformation
->SetColumnItems(2, &ColumnTitles[2], 1, -1, CATDlgDataAdd);
        _MultiListInformation
->SetColumnItems(3, &ColumnTitles[3], 1, -1, CATDlgDataAdd);

    }

```

5.6.9 PushButtonDelete 响应函数

找到

```

void BITCreateAssemSndDlg::OnPushButtonDeletePushBActivateNotification(CATCommand*
cmd, CATNotification* evt, CATCommandClientData data)
{

```

在大括号中添加:

```

    ////////////////////////////////////// 找到那个被选中的行，并将行号
    值赋给RowNumber，删除行
    CATBool RowIsSelected;
    int RowNumber =0;
    for(int i=0;i<_MultiListInformation->GetLineCount();i++)
    {
        RowIsSelected = _MultiListInformation->IsSelect(i);           //找到那个被选中
    的行
        if(RowIsSelected ==TRUE)  RowNumber = i;
    }

    _MultiListInformation->ClearLine(RowNumber);           //删除行

```

5.6.10 WindClose 响应函数

必须要在前边添加过 WindClose 的回调函数，否则找不到下边句子。

找到

```

void BITCreateAssemSndDlg::OnBITCreateAssemSndDlgWindCloseNotification(CATCommand*
cmd, CATNotification* evt, CATCommandClientData data)
{

```

在大括号中添加:

```

    //////////////////////////////////////关闭对话框
    this->SetVisibility(CATDlgHide);
    this->RequestDelayedDestruction();

```

5.6.11 CATDlgFile 使用

在h文件中声明CATDlgFile *pDlgFile; 和 void ActOnOK(CATCommand * cmd, CATNotification* evt , CATCommandClientData data);

在cpp文件中添加代码:

```
#include"iostream.h"
#include"fstream.h"
#include"CATNotification.h"
void
```

```
BITCreateAssemSndDlg::OnBITCreateAssemSndDlgDiaOKNotification(CATCommand* cmd,
CATNotification* evt, CATCommandClientData data)
```

```
{
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOKCancel);//弹出询问消息
    _OpenNotify->SetText("导出信息? ");
    _OpenNotify->SetVisibility(CATDlgShow);
    pDlgFile = new CATDlgFile(this, "导出", CATDlgFileSave);//弹出文件选择框
    pDlgFile->SetVisibility(CATDlgShow);
    CATUnicodeString nameExtension = CATUnicodeString("txt files");//设置文件格式为txt
    CATString filterExtension = CATString("*.txt");
    pDlgFile->SetFilterStrings(&nameExtension, &filterExtension, 1);
    pDlgFile->SetFileName("信息记录");//设置文件默认名称为信息记录
    int iTypeOfInput = 0;
    //添加文件选择框确定按钮事件回调函数ActOnOK
    AddAnalyseNotificationCB (pDlgFile,
        pDlgFile->GetDiaOKNotification(),
        (CATCommandMethod)&BITCreateAssemSndDlg::ActOnOK,
        &iTypeOfInput);
}
```

//事件回调函数ActOnOK的实现

```
void BITCreateAssemSndDlg::ActOnOK(CATCommand * cmd, CATNotification* evt ,
CATCommandClientData data)
```

```
{
    CATUnicodeString fileName;
    pDlgFile->GetSelection(fileName);//获取选择的路径
    _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyOK);//选择的路径提示框
    _OpenNotify->SetText("文件将被保存到:"+fileName);
    _OpenNotify->SetVisibility(CATDlgShow);

    fileName=fileName+".txt";
    constchar *filepath=fileName.ConvertToChar();
    ofstream outfile(filepath, ios::out|ios::trunc);//用c++的输出文件
    if(!outfile)
    {
        _OpenNotify = new CATDlgNotify(this, "", CATDlgNfyError);
    }
}
```

```

        _OpenNotify->SetText("打开出错!");
        _OpenNotify->SetVisibility(CATDlgShow);
    }
    outfile<<"姓名 "<<"性别 "<<"年龄 "<<"喜欢学科  "<<endl;
    CATUnicodeString name, gender, age, subject;
    int ColumnNum=_MultiListInformation->GetLineCount();
    for(int i=0;i<ColumnNum;i++)
    {
        //获取MultiListInformation中的信息
        _MultiListInformation->GetColumnItem(0, name, i);
        _MultiListInformation->GetColumnItem(1, gender, i);
        _MultiListInformation->GetColumnItem(3, age, i);
        _MultiListInformation->GetColumnItem(2, subject, i);
        //此处需要调整字符串的输出方式,使得txt格式整齐
        name.Resize(6, ' ', 0);
        gender.Resize(4, ' ', 0);
        age.Resize(4, ' ', 0);
        outfile<<name<<" "<<gender<<" "<<age<<" "<<subject<<" "<<endl;
    }
    pDlgFile->RequestDelayedDestruction();
    pDlgFile=NULL;
}

```

以上代码完成记录保存成为txt文件。

5.6.12 MultiListInformation 的选择响应函数

void

BITCreateAssemSndDlg::OnMultiListInformationListSelectNotification(CATCommand* cmd, CATNotification* evt, CATCommandClientData data)

```

{
    int selectLine[1];
    int selectNumb = _MultiListInformation->GetSelect(selectLine, 1);
    CATUnicodeString name, gender, age, subject;
    _MultiListInformation->GetColumnItem(0, name, selectLine[0]);
    _MultiListInformation->GetColumnItem(1, gender, selectLine[0]);
    _MultiListInformation->GetColumnItem(3, age, selectLine[0]);
    _MultiListInformation->GetColumnItem(2, subject, selectLine[0]);
    _EditorName->SetText(name); //Editor值设置
    if(gender=="男")
    {_RadioButtonMale->SetState(CATDlgCheck);_RadioButtonFemale->SetState(CATDlgUncheck);} //RadioButton状态设置
    if(gender=="女")
    {_RadioButtonMale->SetState(CATDlgUncheck);_RadioButtonFemale->SetState(CATDlgCheck);}

    int ageNumb;
}

```

```

age.ConvertToNum(&ageNumb, "%d");//字符串转化为整数
_ComboAge->SetSelect(ageNumb-20);
if(-1==subject.SearchSubString("知识工程"))//字符串搜索
    _CheckButtonKnowledge->SetState(CATDlgUncheck);//CheckBox状态设置
else _CheckButtonKnowledge->SetState(CATDlgCheck);
if(-1==subject.SearchSubString("设计制造集成"))
    _CheckButtonDesign->SetState(CATDlgUncheck);
else _CheckButtonDesign->SetState(CATDlgCheck);
if(-1==subject.SearchSubString("三维工艺设计"))
    _CheckButton3D->SetState(CATDlgUncheck);
else _CheckButton3D->SetState(CATDlgCheck);
}

```

5.7 完整实例运行

运行程序：

- A. CAAV5 Workspace->Create/Update Runtime View->Ok
- B. 生成->mkmk->OK
- C. 窗口->Open runtime Window-输入"Cnext"->回车
- D. 装配 a->信息录入

6 总结

学习过程中必须注意以下几点：

A. 在 CAA 开发过程中很大一部分代码是自动生成的。一般不要轻易修改自动生成的代码。

B. 文件->add CAA V5 Item-> CATIA resource->dialog 方式添加的对话框。 .h 文件中已经声明了所有控件的指针，不需要自己声明。自己只需要声明：CATDlgNotify

C. 所有用到的函数都可以在 CAA V5 Help Viewer 中查到。打开方式为：社区->Help CAA V5。

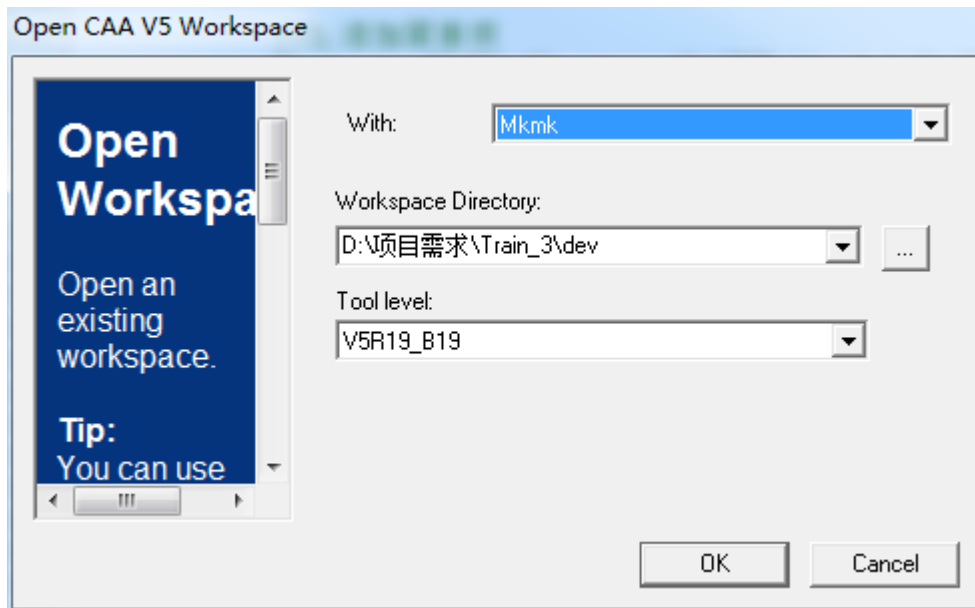
比如要查找与类 CATDlgEditor 相关的函数就在 Index 选项卡中输入 CATDlgEditor。

D. 与对话框中控件的操作相关的文档在百科全书里边查找：

User Interface ->WinTop Dialogs

E. 刚开始编程程序出错可能的原因有：头文件是否添加完全、变量是否声明、当界面更改时是否更新

F.所有的 Train 附带的 Case 都能使用，方法为：打开 Visual Studio->文件->Open CAAV5 Workspace 弹出对话框：



选择 Workspace Directory: xxxx\Train_x\xx

其他选项按照上图选择。点击 OK 打开即可查看源代码。

若要运行，则首先需要 Locate prerequisite

G. 很多东西需要不断的尝试，探索，所以遇到难题不要灰心。

H. 有些头文件放在.h 文件和.cpp 文件里是完全不一样的情况，所以当把头文件的位置改变，可能会导致出错。

I. .CATDllg 文件的编辑器很脆弱，一不小心就会挂掉，所以在编辑对话框的时候要仔细、小心。否则只能重启电脑了（系统内部有个服务会崩溃掉，重启之后又会打开）。

J. 百科全书是一系列的脱机网页文件：目录为：

安装目录\B19\CAAV5HomePage

打开就可以使用。