计算动力学程序设计平台 DynGui
Computational Dynamics Platform

张 雄

2004年11月

计算动力学程序设计平台 DynGui

- 设计环境
- Windows环境下程序设计方法
- 程序界面
- 程序结构
- 混合语言编程
- 程序设计接口

  菜单、对话框、提示窗口、命令、工具栏
  位图按钮、动画显示、OpenGL图形

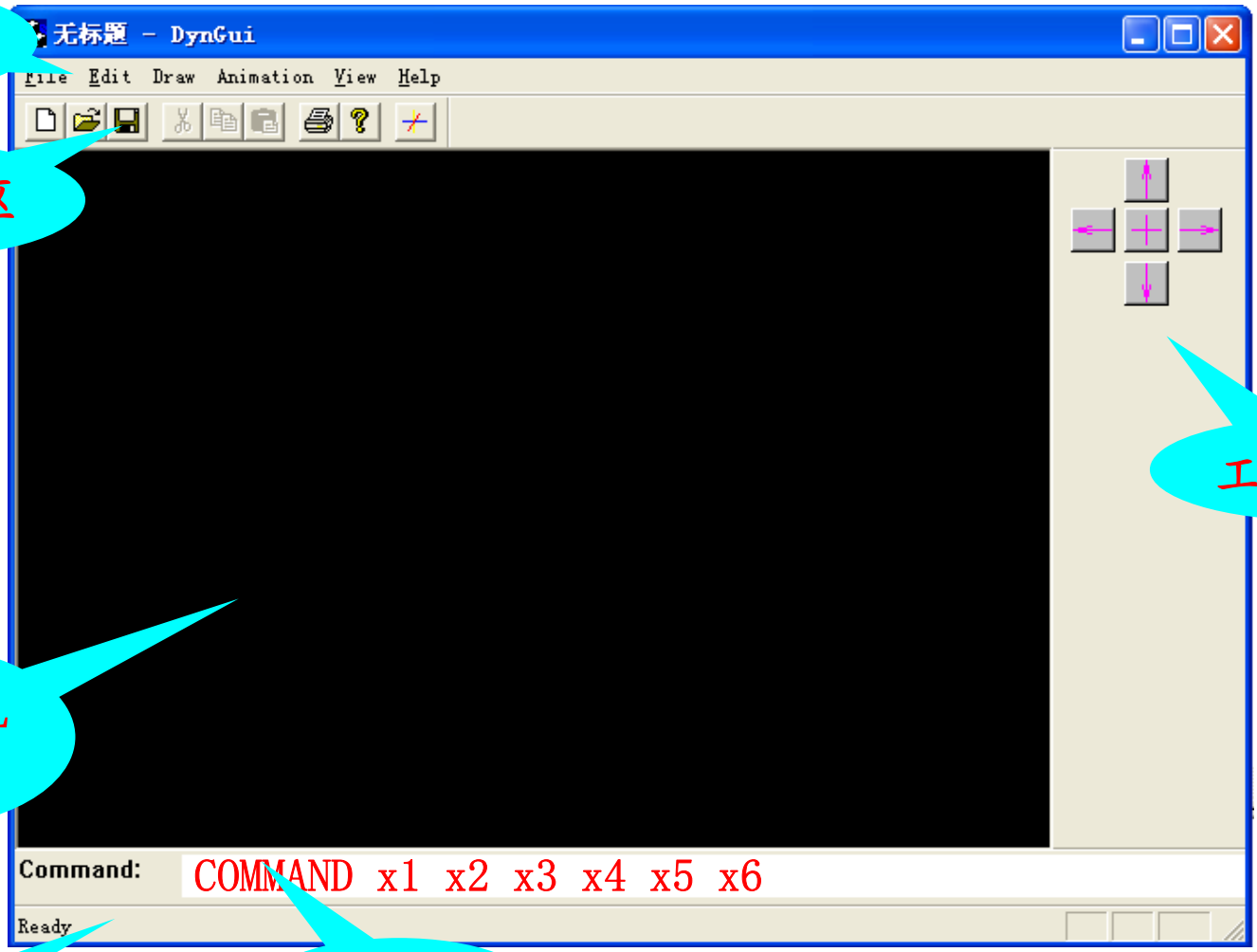
设计环境

- Windows操作系统
- Visual C++ 6.0 + FORTRAN + C
- Microsoft基础类MFC (Microsoft Foundation Class) — 开发Windows环境下的GUI应用程序
- OpenGL — 三维交互式图形处理

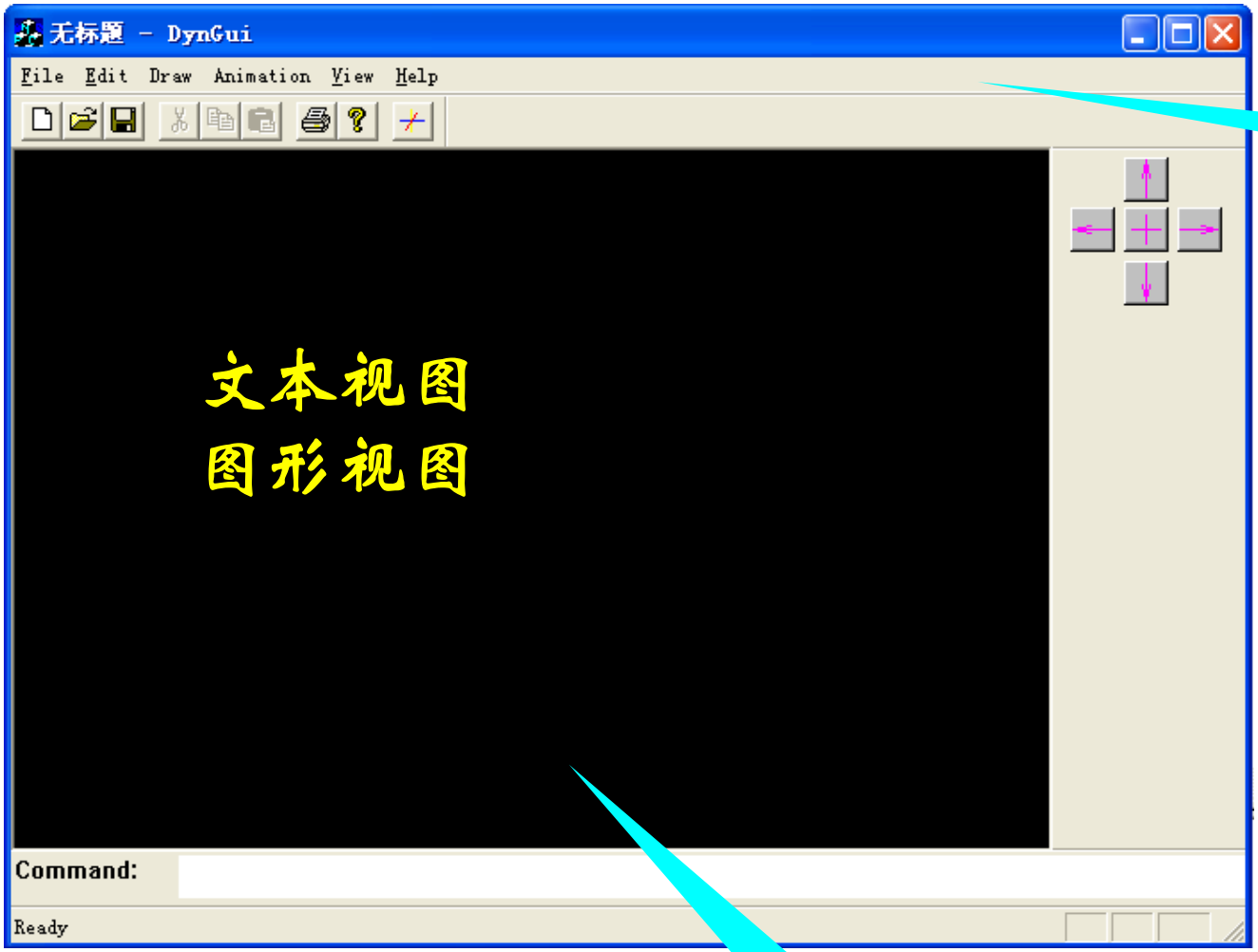
Windows环境下程序设计方法

- 事件（消息）驱动 — 编写事件处理函数
- 图形用户界面 — 菜单、对话框、控件
- 使用ClassWizard管理类

程序界面 — 单文档界面(SDI)



程序结构 — 文档/视图结构


无标题 - DynGui
File Edit Draw Animation View Help
文本视图
图形视图
Command:
Ready
框架类
视图类

程序结构

- 程序类 CDynGuiApp
- 文档类 CDynGuiDoc
- 框架类 CMainFrame
- 视图类 —两个视图间可相互切换
  - CDynGuiView : CGLView : CView
    OpenGL图形视图，显示三维图形
  - CDynScrollView : CView
    文本视图，显示计算结果等文本信息

程序结构

- 对话框类 — 交互式输入/输出
  - 多边形填充方式：CPolygonFillDlg
  - 模型放大系数： CScaleDlg
  - 状态号：CStateNoDlg
  - 转动角度增量：CStepSizeDlg
  - 切换视图号：CSwitchToDlg
  - 动画显示时间间隔：CTimeStepDlg
  - 模型平移量：CTranslateDlg
  - 工具盒：CToolDialogBar
  - 关于：CAboutDlg

程序结构

- 命令处理：CCmdRun
- 随机文件存取(缓冲、文件族)：CFFRandIO
- 有限元结果可视化(基类)：CFEModel
- DYNA3D可视化类：CPostDyna: CFEModel
  - ImportFEModel(char* FileName)
  - LoadNodalState(int state)
  - LoadBrickEleState(int state)
- OpenGL演示：CGLDemo
- 行文本输出：CLinePrint

只定义与**DYNA3D**结果文件结构相关的函数，其他所有函数由基类**CFEModel**定义

混合语言编程

- C++调用FORTRAN

  Calling conventions; Naming conventions;

  Passing by value or by reference

  - 约定差异
    - 数组存储：按行(C)或按列(FORTRAN)
    - 实亚结合：值传递(C)、地址传递(FORTRAN)
    - 字符串传递：串地址+串长度(FORTRAN)
  - 调用方法
    - "C"：_name@nn (C++：_name@@decoration)
    - __stdcall：逆序入栈、_name@nn

混合语言编程

- C++调用FORTRAN
  - C++

```
extern "C"
{
    void __stdcall RWABSF(int* FIT, char* FN,
        int LFN, int* US, int* BFS, int* FCS);
}
```

  - FORTRAN

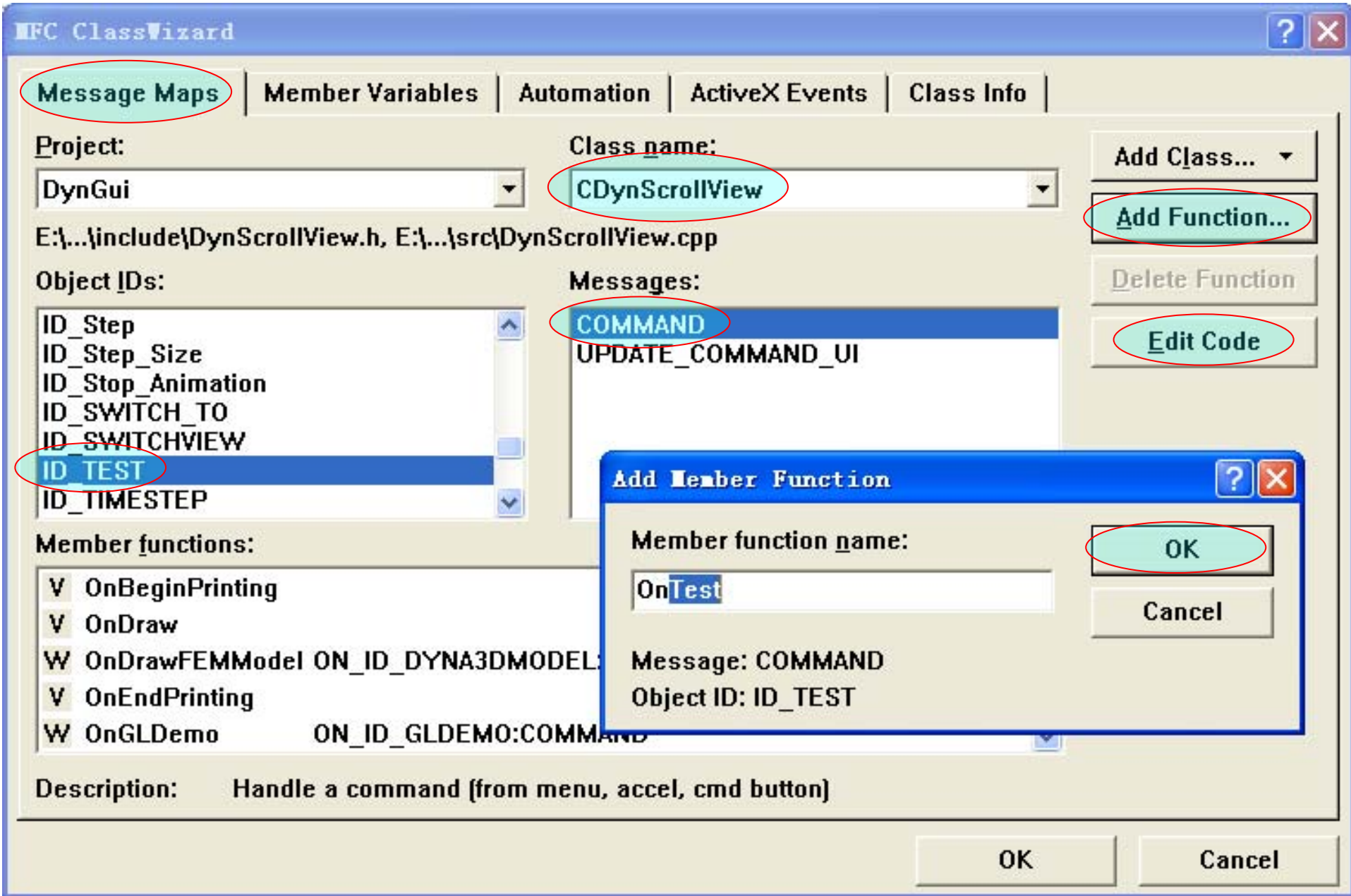
```
SUBROUTINE RWABSF (FIT,LFN,US,BFS,FCS)
```

- 实例： CFFRandIO

程序设计接口 添加菜单

- Resource/Menu/IDR_MAINFRAME
  - Caption (TEST)
  - ID (ID_TEST)
- Where to response ID_TEST message ?
  - CMainFrame – Available to all views
  - CDynGuiView – Only available to this view
  - CDynScrollView

- ClassWizard



程序设计接口　　添加菜单

- DynScrollView.h

  afx_msg void OnTest();　函数原型说明

- DynScrollView.CPP

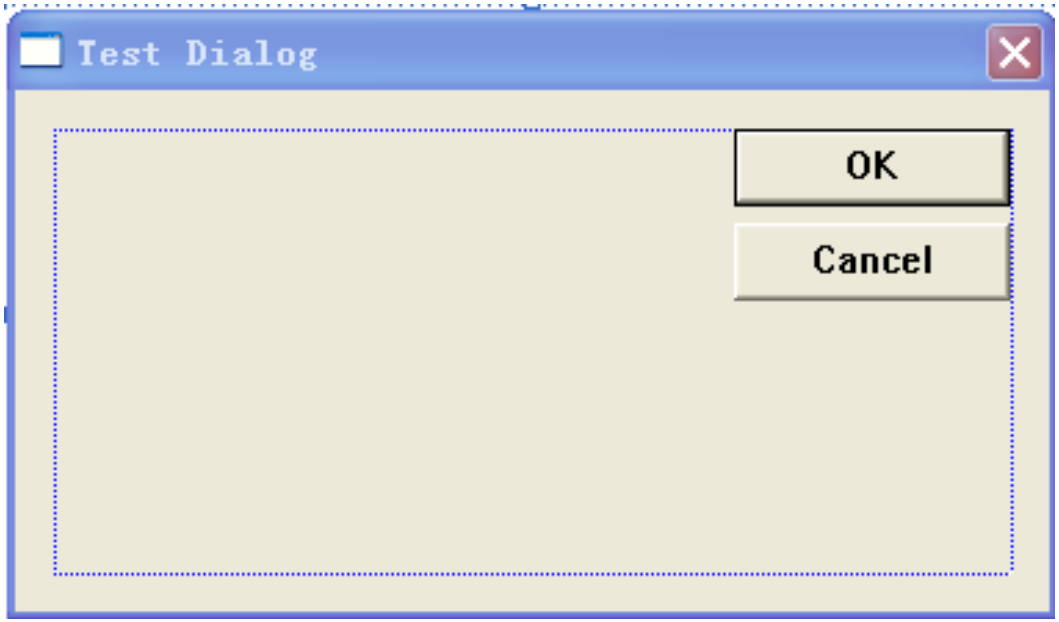
  ON_COMMAND(ID_TEST, OnTest)　定义映射关系

  void CDynScrollView::OnTest() 定义消息处理函数

  ```
  {
      // TODO: Add your command handler code here
  }
  ```

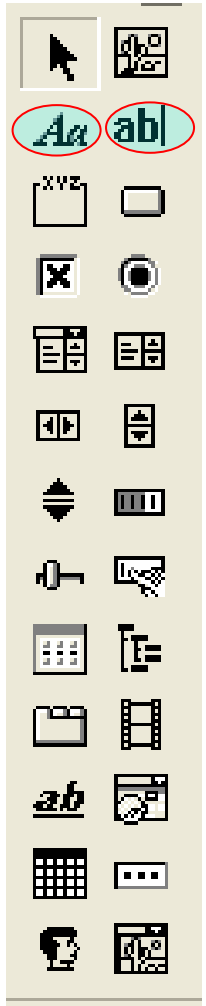
程序设计接口　　添加对话框

- Resource/Dialog/Insert Dialog/



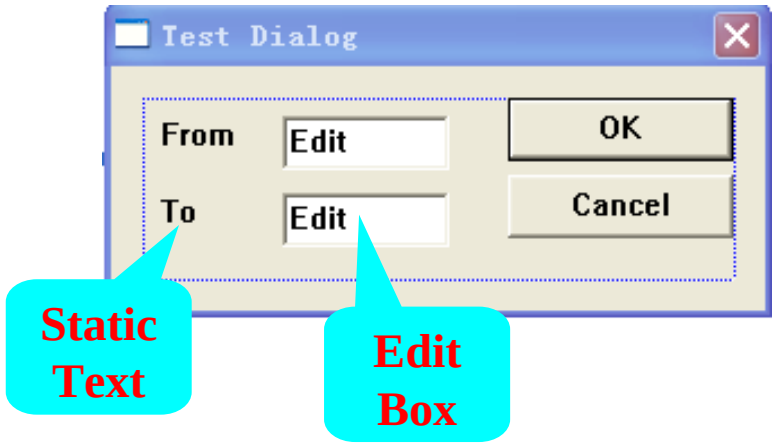
Caption: Test Dialog

ID: IDD_TEST_DIALOG

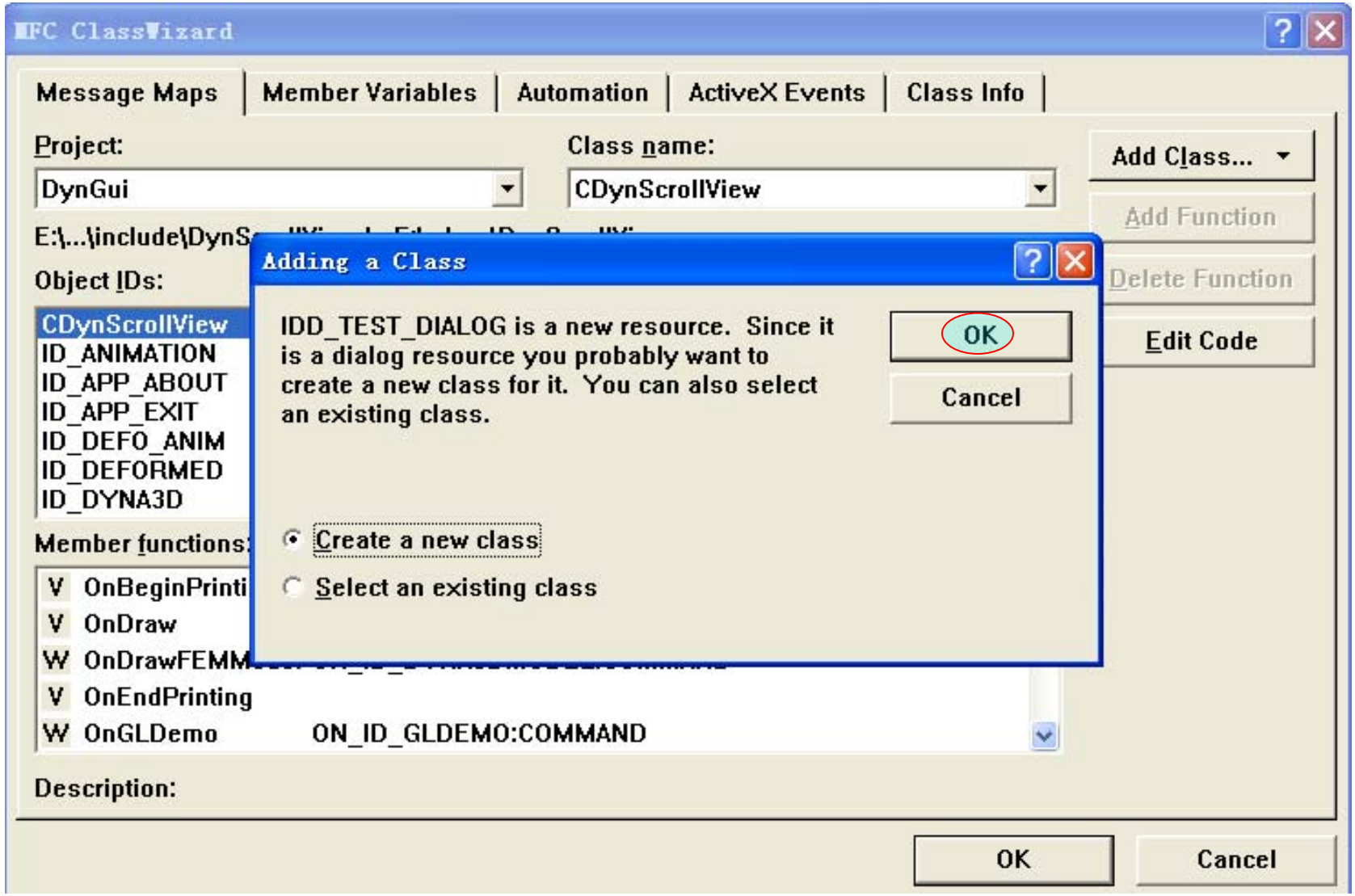


程序设计接口　　添加对话框

- 增加Controls

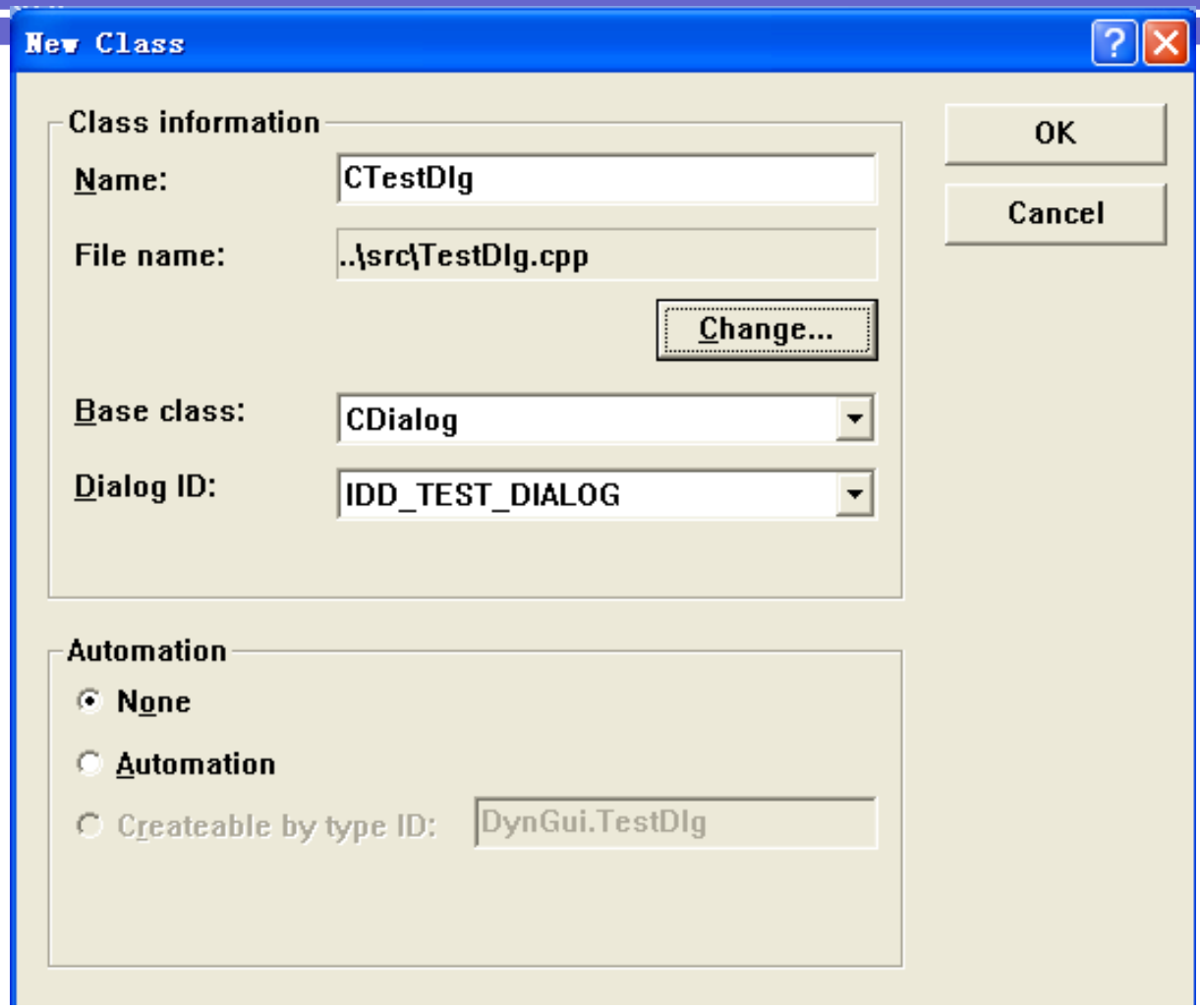


- Create a new class

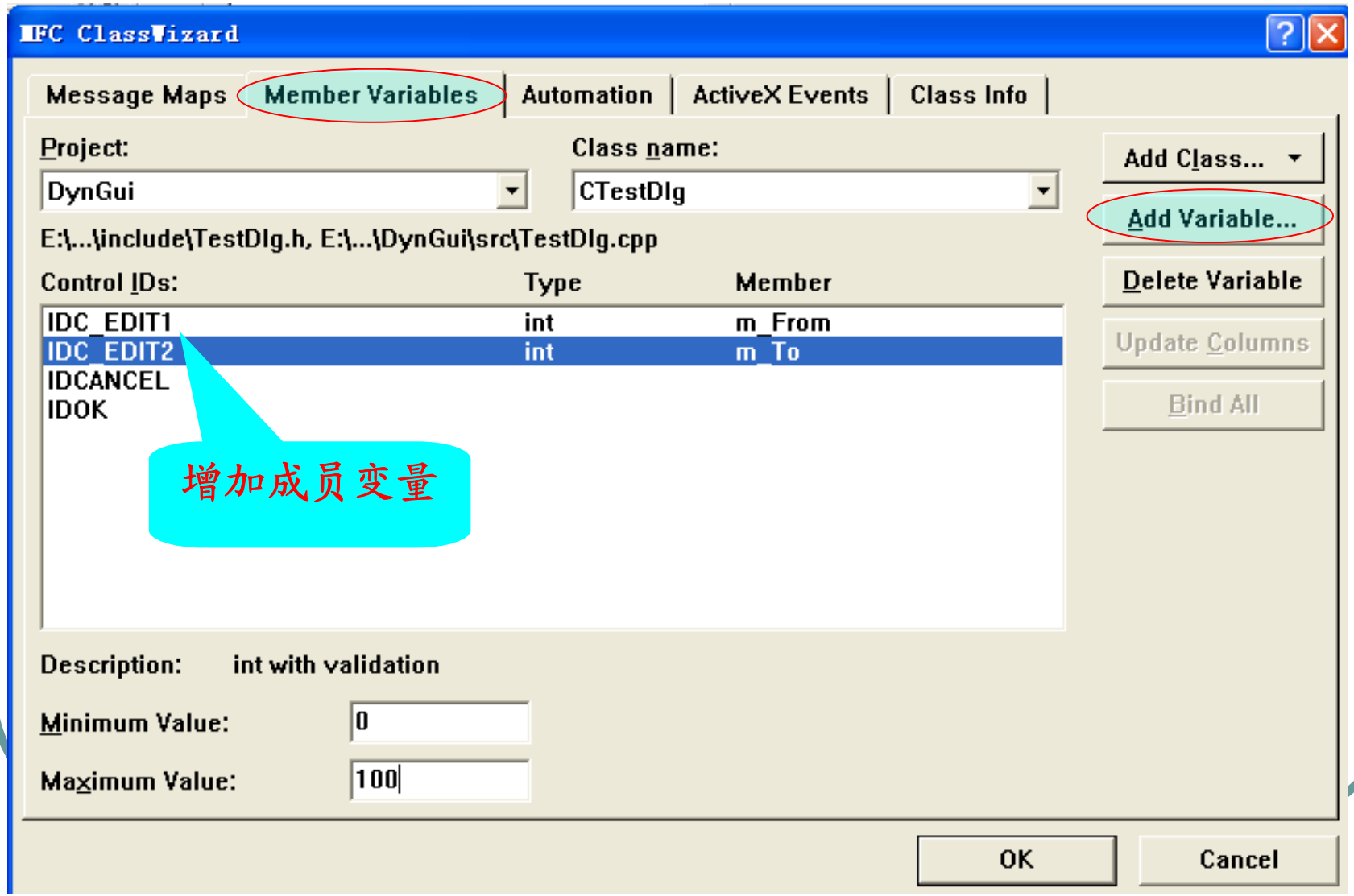


程序设计接口

添加对话框



程序设计接口　　添加对话框



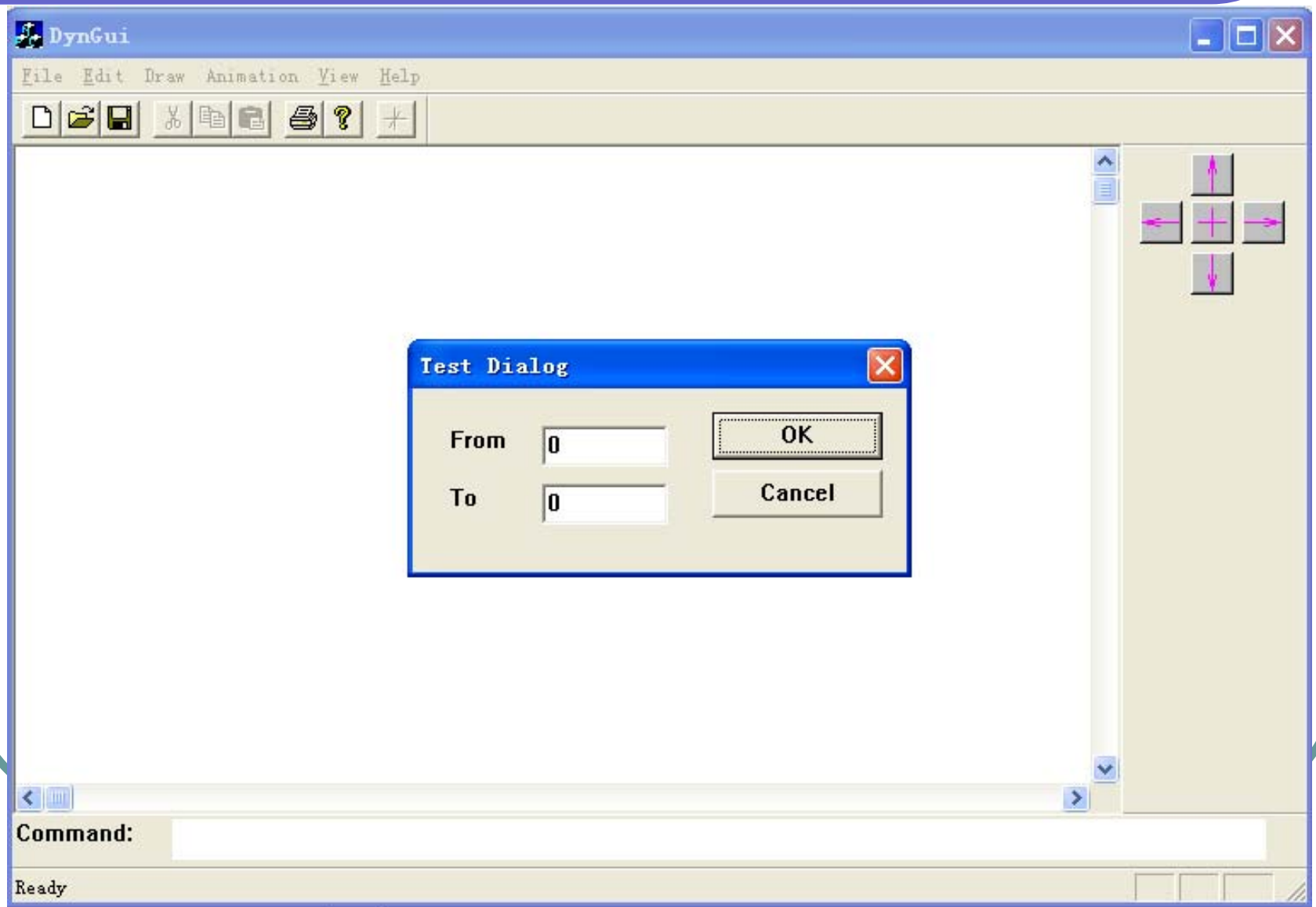
程序设计接口　　使用对话框

```
void CDynScrollView::OnTest()
{
   CTestDlg TestDlg;
   TestDlg.m_From = m_From;
   TestDlg.m_To   = m_To;
   TestDlg.DoModal();
   m_From = TestDlg.m_From;
   m_To   = TestDlg.m_To;
}
```

类成员变量

程序设计接口

运行效果


DynGui
File Edit Draw Animation View Help
Test Dialog
From
0
To
0
OK
Cancel
Command:
Ready

程序设计接口　　提示窗口

- C

```
AfxMessageBox("Error: No free unit available.");
```

- FORTRAN

```
USE DFWIN
INF = MessageBox(0, 'I/O ERROR!',"Error", MB_OK)
```



程序设计接口　　添加命令

- CmdRun.h:

```
const int m_nParameters = 6; (命令参数最大数)
const int m_NumKeyword  = 21;(命令总数)
const int m_NumHelpText = 50;(帮助文本行数)
```

- CmdRun.cpp

```
CString CCmdRun::KeyWords[m_NumKeyword] = {
    "SVIEW$", "DRAW$", "ROTA$TE", "RX$", …… 
};
CString CCmdRun::HelpText[m_NumHelpText] = {
    "SVIW - Switch views ", …… 
}
```

增加命令

增加命令
帮助文本

程序设计接口　　添加命令

- 在Process中增加命令处理程序段

```
switch (nKeyWord(cmd, cPara, mPara, m_nParas))
{
  case 0 : // SVIEW : Switch views
    pMainFrm->OnSwitchView();
    break;
    … …
  case 19:
    pDynGui->OnUndeformed();
    pDynGui->Invalidate();
    break;
    … …
}
```

字符型参数

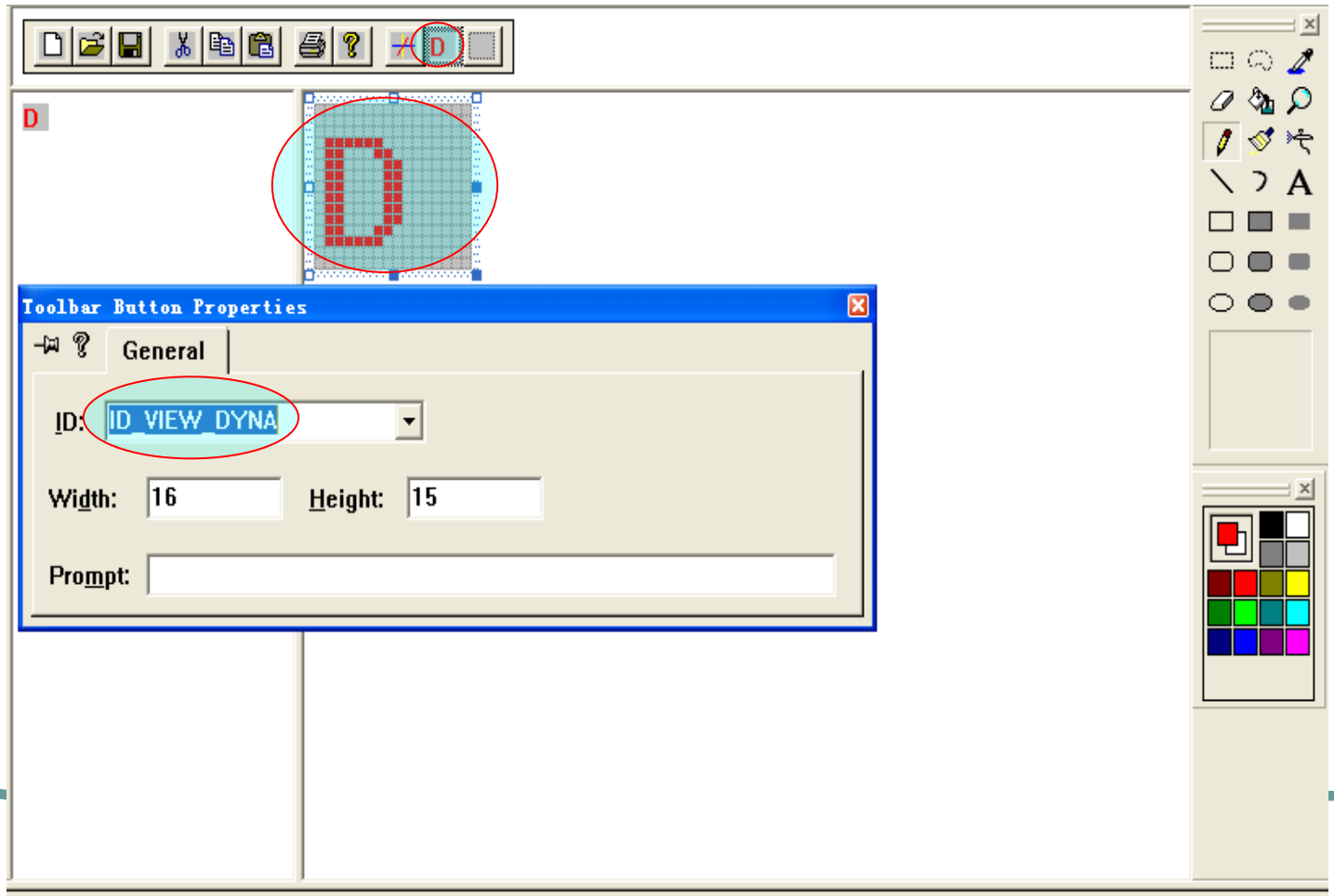
数值型参数

参数总数

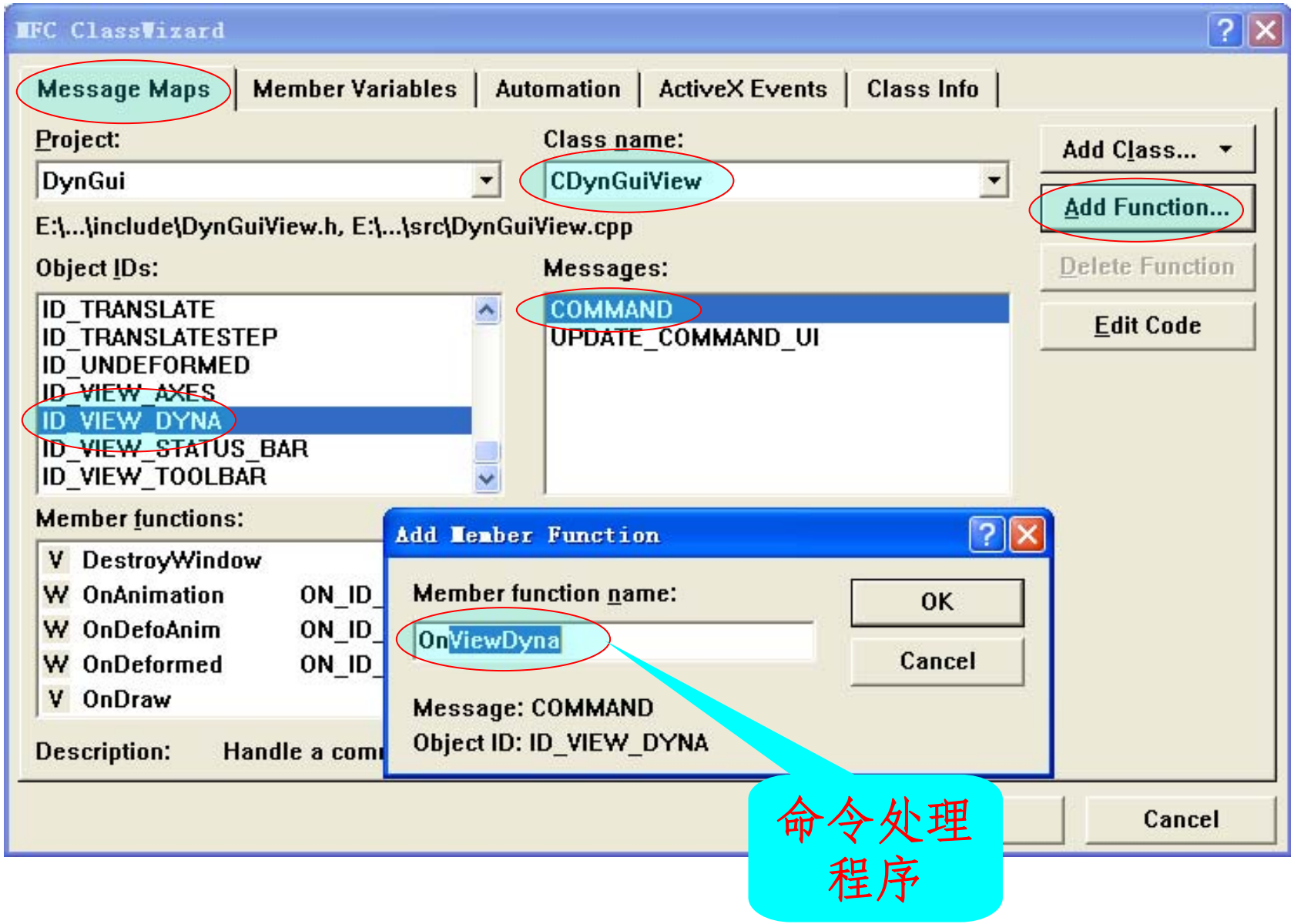
增加命令处理程序段

程序设计接口　　添加工具栏

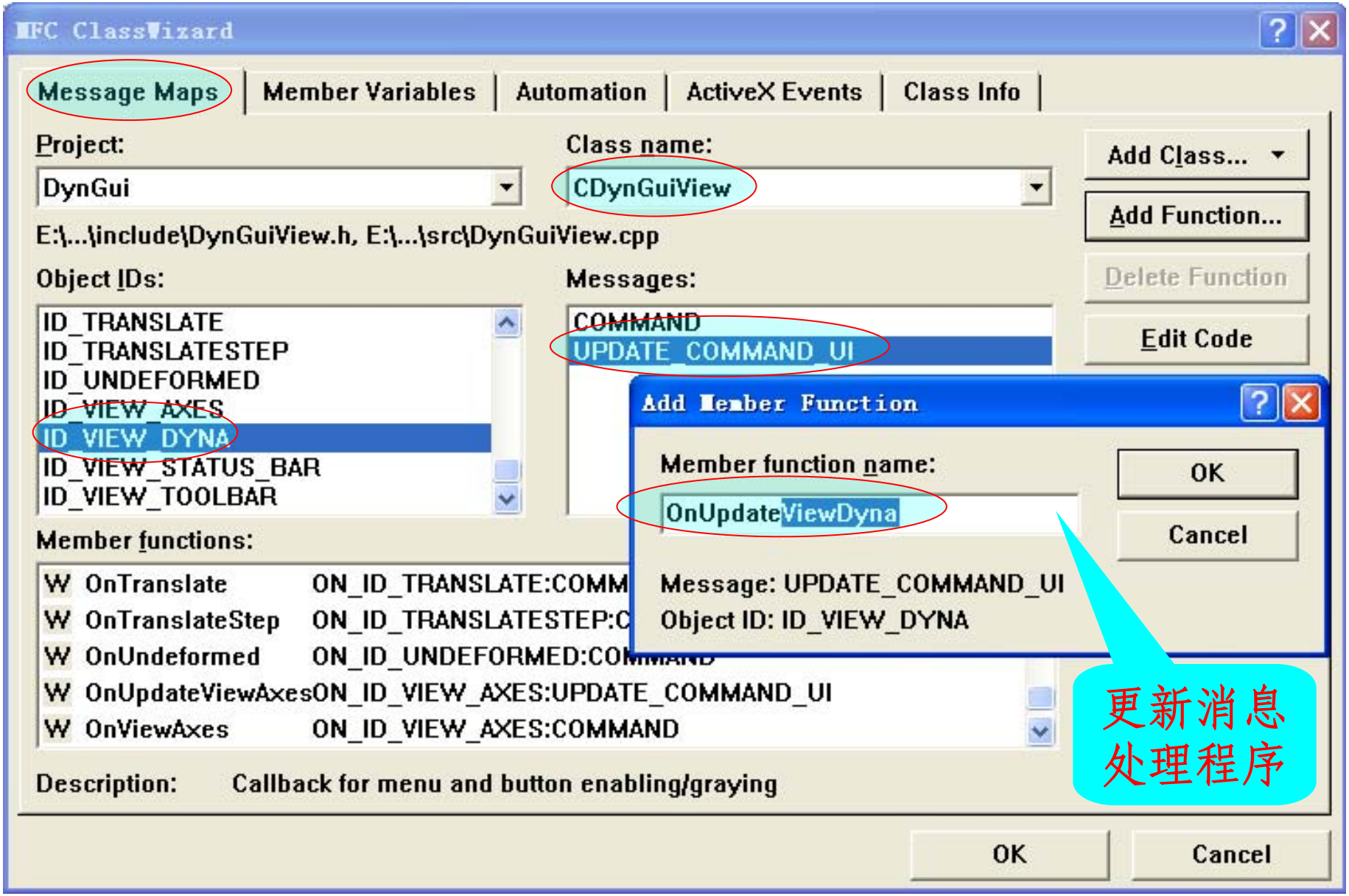
- Resourse/Toolbar/IDR_MAINFRAME



- ClassWizard



- ClassWizard

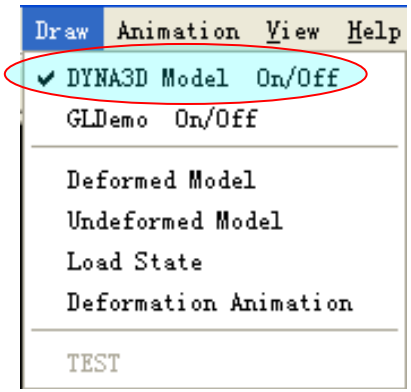


程序设计接口　　添加工具栏

```
void CDynGuiView::OnUpdateViewDyna(CCmdUI* pCmdUI)
{
    // Check/uncheck the option
    if(p_FEModel)
       pCmdUI->SetCheck(b_ViewFEModel);
}
```

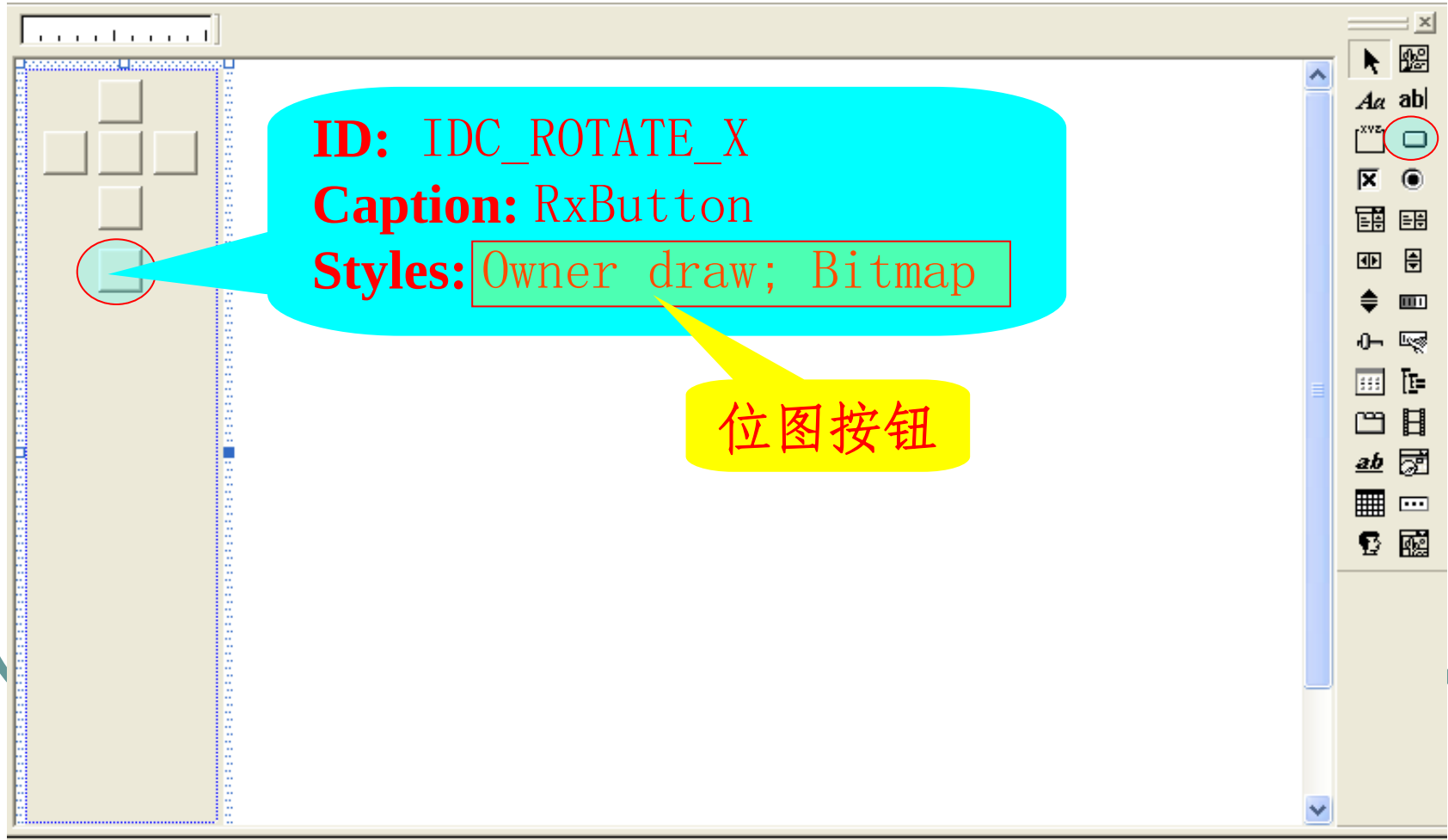






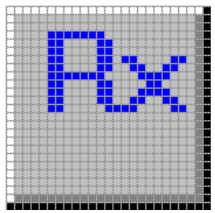
程序设计接口　　添加位图按钮

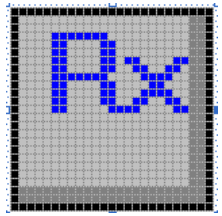
- Resourse/Dialog/IDD_TOOL_DIALOG

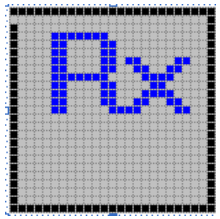


程序设计接口　　添加位图按钮

- Resourse/Bitmap/Insert Bitmap

“RXBUTTONU”  **Up**

“RXBUTTONF”  **Get focus**

“RXBUTTOND”  **Down**

Only for Bitmap button

程序设计接口　　添加位图按钮

- ToolDialogBar.h

```
CBitmapButton m_LeftButton;
… …
CBitmapButton m_RxButton;
```

定义位图按钮

- ToolDialogBar.cpp

```
void CToolDialogBar::InitDialog()
{
  m_LeftButton.AutoLoad(IDC_LEFT_BUTTON,this);
  … …
  m_RxButton.AutoLoad(IDC_ROTATE_X,this);
}
```

载入位图按钮

按钮ID

Only for Bitmap button

程序设计接口　　添加位图按钮

- MainFrm.h

```
afx_msg void OnLeftButton();
… …
afx_msg void OnRxButton();
```

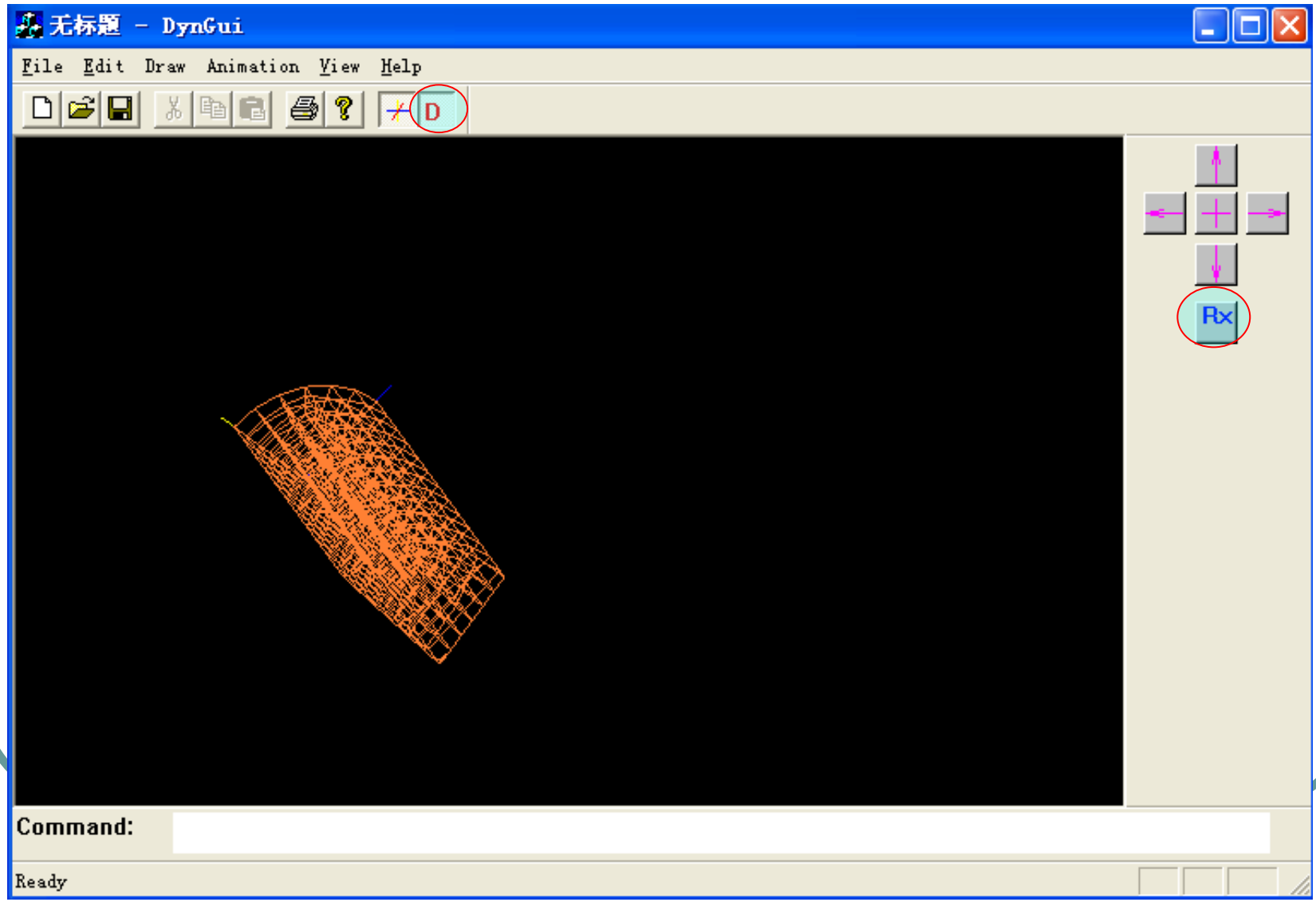
消息处理函数

- MainFrm.cpp

```
ON_COMMAND(IDC_LEFT_BUTTON, OnLeftButton)
… …
ON_COMMAND(IDC_ROTATE_X, OnRxButton)

void CMainFrame::OnRxButton()
{
   cmd.Format("RX");
   SplitPartsString(cmd, s_cmdParts);
   CmdRun.Process(this,s_cmdParts,this->GetActiveView());
}
```

程序设计接口　　运行效果


无标题 - DynGui
File Edit Draw Animation View Help
D
Rx
Command:
Ready

程序设计接口　　动画显示

- 模型动画

```
void CDynGuiView::OnAnimation()
{
   m_CurrentTimer = GUITIMER;  // Set up a timer
   m_iTimerID = SetTimer(m_CurrentTimer,
   m_TimeStepSize, NULL);
}
```

设置Timer，每隔 m_TimeStepSize (ms) 调用一次 OnTimer

程序设计接口　　动画显示

- 有限元结果动画

```
void CDynGuiView::DeformedAnim()
{
   p_FEModel->CurrentState(m_FirstState);
   m_CurrentTimer = FEMTIMER;
   m_iTimerID = SetTimer(m_CurrentTimer,
   m_TimeStepSize, NULL);
}
```

设置**Timer**，每隔 **m_TimeStepSize (ms)** 调用一次 **OnTimer**

程序设计接口　　动画显示

```
void CDynGuiView::OnTimer(UINT nIDEvent)
{
   if (nIDEvent == GUITIMER) {
       m_fXRotate += m_RxStepSize;
       m_fYRotate += m_RyStepSize;
       m_fZRotate += m_RzStepSize;
   }
   else if (nIDEvent == FEMTIMER) {
       p_FEModel->LoadCurrNodalState();
       p_FEModel->NextState(m_StateInc);
   }
   Invalidate();   // Redraw
   CGLView::OnTimer(nIDEvent);
}
```

```
void CDynGuiView::OnDraw(CDC* pDC)
{
  CDynGuiDoc* pDoc = GetDocument();
  wglMakeCurrent(pDC->m_hDC,m_hRC);
      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
      glMatrixMode(GL_MODELVIEW);
      glPushMatrix();
      glRotatef(m_fXRotate,1.0f,0.0f,0.0f);
      glRotatef(m_fYRotate,0.0f,1.0f,0.0f);
      glRotatef(m_fZRotate,0.0f,0.0f,1.0f);
      glScalef(m_Scale,m_Scale,m_Scale);
      glTranslatef(m_TranslateX,m_TranslateY,m_TranslateZ);
      if (b_ViewFEModel && p_FEModel)
            p_FEModel->RenderStruView();
      if (b_ViewGLDemo && p_GLDemo)
            p_GLDemo->RenderView();
      glPopMatrix();
      SwapBuffers(pDC->m_hDC);
  wglMakeCurrent(pDC->m_hDC,NULL);
}
```

程序设计接口 OpenGL图形

- 准备工作：由GLView类自动完成
  - OnCreate：建立硬件连接
  - OnSize：设置视窗大小和观察方式
- 绘图工作：在OnDraw中进行
  - 模型变换 — 平移、旋转
  - 模型绘制 — 充分利用显示列表（参见函数CFEModel :: RenderStruView()）

现有命令

```
SVIEW
DRAW [DYNA/GLDEMO]
ROTA [fx] [fy] [fz]
RX [fx]  |  RY [fy]  |  RZ [fz]
SCAL s
ANIM STARt/STOP/STEP/DEFO/TIME/RX/RY/RZ
HELP
TRAN [fx] [fy] [fz]
TX [fx]  |  TY [fy]  |  TZ [fz]
PROJ ORTH/PERS
FILL/LINE/POINT
TC  |  RC
STAT nst
DEFO [nFirst] [nLast] [nInc]
UNDE
```

To Do List

- 鼠标消息处理
- 新位图按钮
- 新命令
- 应力云图及等值线
- 其他程序（无网格、物质点）数据
- 结果曲线
- … …