

1996

## **TestReview 1.0: User's Manual**

John Chituanya Okoli

Follow this and additional works at: [https://digitalcommons.lsu.edu/honors\\_etd](https://digitalcommons.lsu.edu/honors_etd)



Part of the [Education Commons](#), and the [Engineering Commons](#)

---

# ***TestReview<sup>TM</sup> 1.0***

## **User's Manual**

---

## **Table of Contents**

<b>Introduction</b>	<b>2</b>
<b>Compatibility</b>	<b>2</b>
<b>Overview of the TestReview Views</b>	<b>3</b>
The Design View	3
The Test View	3
<b>Detailed Description of TestReview 1.0 Features</b>	<b>4</b>
Main Window	4
Design View	7
Test View	10
<b>Glossary</b>	<b>12</b>

## Introduction

TestReview is a program designed to let instructors create electronic tests that students can take on a computer. TestReview provides an interface that lets instructors visually create text boxes in which they can type questions or labels, and lets them create other text boxes in which students can respond to questions. This information can be saved to a question file. Students that wish to take the test recorded in a question file can create a test file, which is based on a question file created by an instructor.

This manual documents version 1.0 of TestReview. This version has the basic functionality required to fully enable instructors create electronic tests and let students take these tests.

In this manual, the term “instructor” will refer to the person who creates tests with TestReview and “student” will refer to the person who takes a test created by an “instructor”.

TestReview 1.0 is freeware. It may be used and distributed free of charge on the condition that it is not sold for any purposes whatsoever. TestReview 1.0 is delivered on an as-is basis; there is no warranty, expressed or implied, on the reliability of this product, and there is no support provided.

TestReview consists of one file: TestReview.exe.

## Compatibility

TestReview was written for Win32 applications, so it will work on Microsoft Windows 95, Microsoft Windows NT and compatible operating systems. It will not work on Microsoft Windows systems prior to Windows 95.

The minimum hardware requirements are the same as those for Windows 95:

- Personal computer with 386DX or higher processor
- MB of memory
- MB of hard disk space
- One 3.5-inch floppy disk drive (for installation)
- VGA or higher resolution graphics card

---

## Overview of the TestReview Views

### The Design View

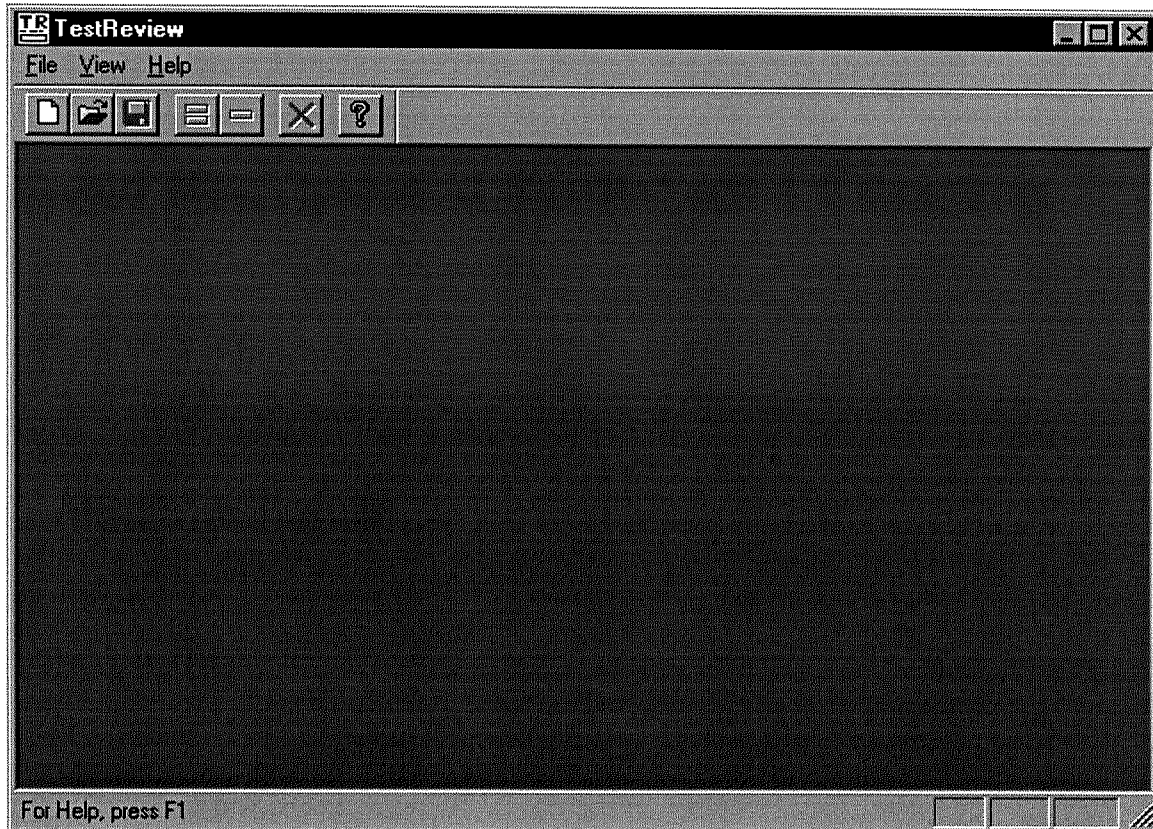
The Design View is used only by instructors. This is the view that permits the creation of tests. Both the menu and the toolbar in the Design View provide commands for adding and deleting new question, response and label text boxes. The Design View is used to create question files. TestReview Question files have the file extension **.TRQ**. A question file must exist before a test file is created in the Test View.

### The Test View


The Test View is used primarily by students, but instructors will also want to use it to see what the question files they create will look like. The Test View allows a student to take a test that is stored in a question file. The Test View creates TestReview Test files; these files have the extension **.TRT**. When a student wants to create a test file in the Test View, the student will be prompted to specify a question (.trq) file upon which the new test file will be based.

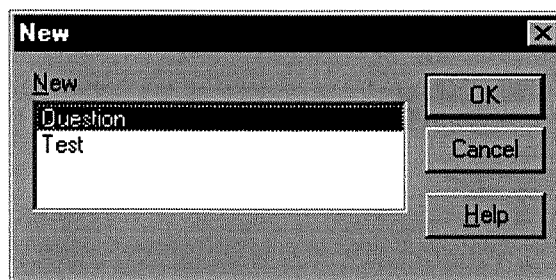
## Detailed Description of TestReview 1.0 Features

### Main Window



### File Menu


 **New:** This command brings up the New file dialog box.

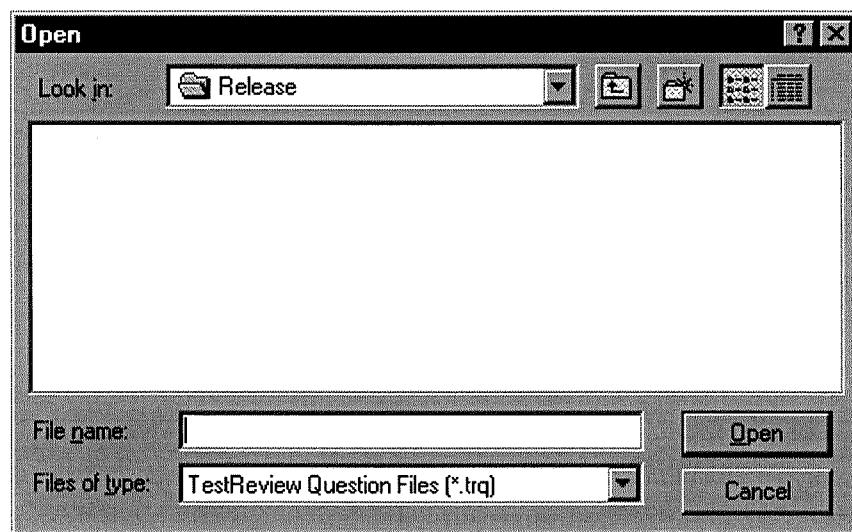


To create a new question file select “Question”, then click “OK”. To work with question files, see the section on **Design View**.

To create a new test file, select “Test”, then click “OK”. Before you can create a new test file, you must select a question (.TRQ) file that the test will be based upon. When you click “OK”, you will see a message box reminding you of this, then you will see the Open file dialog box (see **Open**) which will let you select and open a **question** file. To work with test files, see the section on **Test View**.

Click “Cancel” to cancel.

 **Open:** This command brings up the Open file dialog box.



To open an existing question file, make sure that “Files of type:” specifies question files. Locate the question file you want to open, then click “OK”.

To work with question files, see the section on **Design View**.

To open an existing test file, make sure that “Files of type:” specifies test files.

Locate the test file you want to open, then click “OK”. To work with test files, see the section on **Test View**.

Click “Cancel” to cancel.

**MRU List:** This is a list of the files you **most recently used**. Click on one of the file names to automatically open it.


**Exit:** This command exits TestReview.

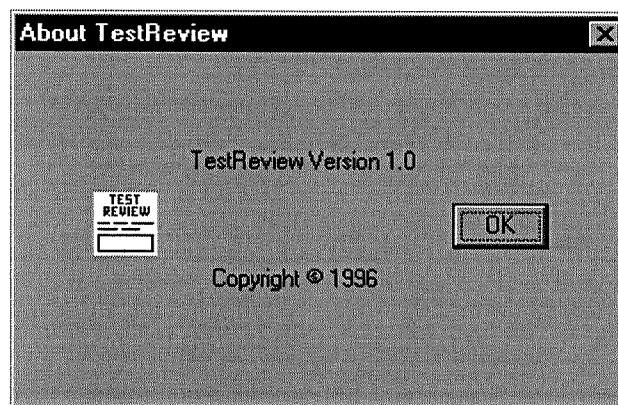
## View Menu

**Toolbar:** When this item is checked, the toolbar is visible.

**Status bar:** When this item is checked, the status bar is visible.

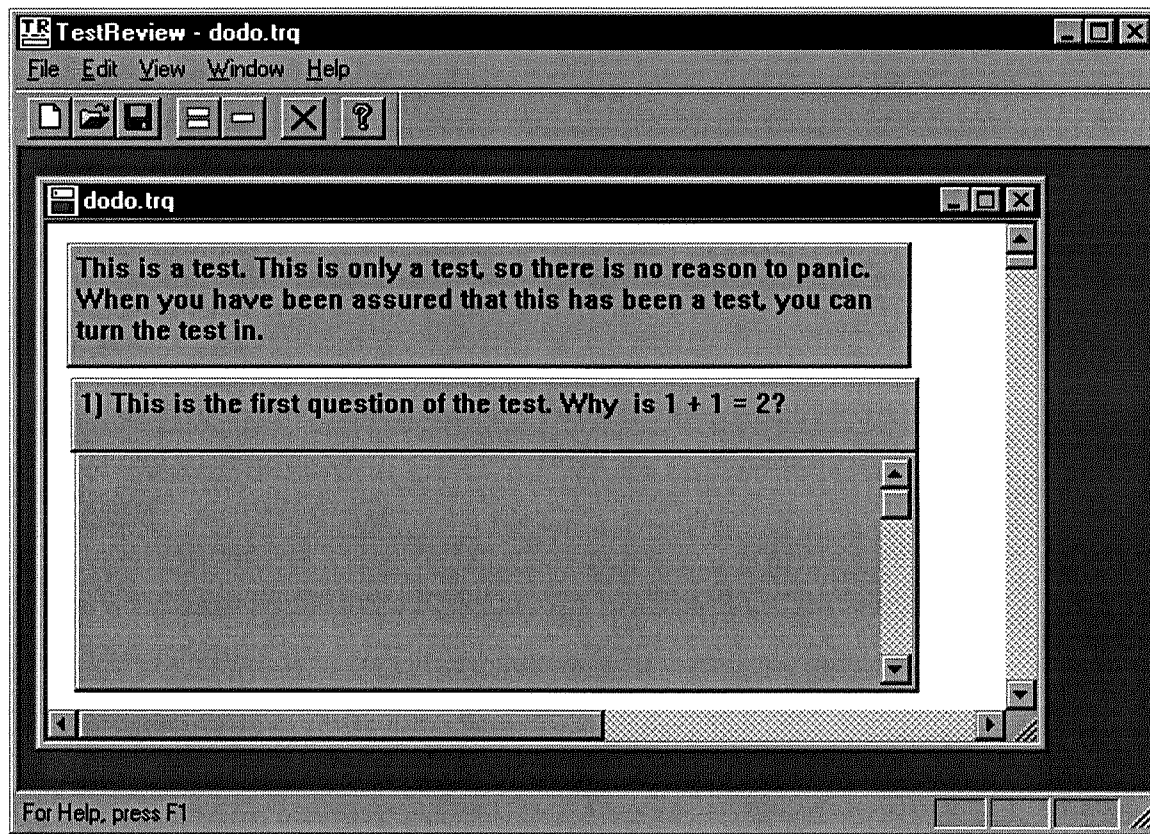
## Help Menu

 **About TestReview:** This command brings up the TestReview About dialog.





## Design View



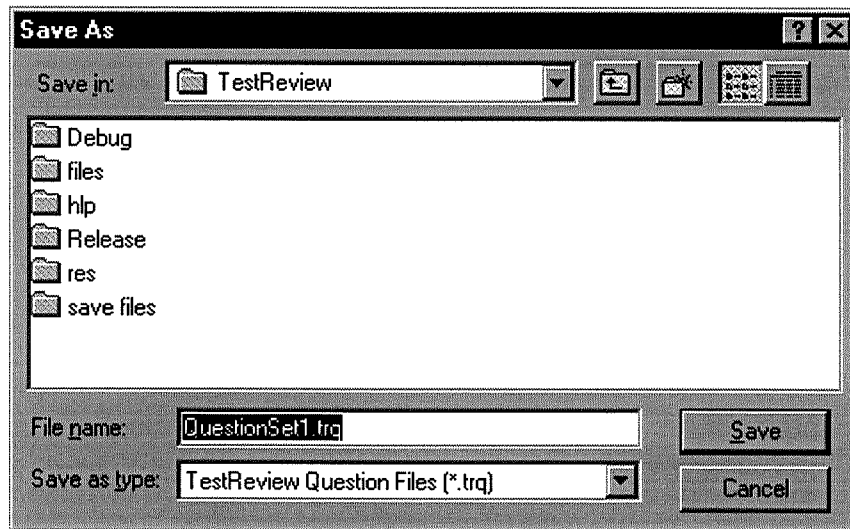
### File Menu

**New**, **Open**, **MRU List**, **Exit**: Same as in **Main Window**.

**Close**: This command closes the currently open file.


**Save**: This command saves the currently open file. If the file has not been saved before, the Save As file dialog box comes up. See **Save As**.


**Save As:** This command saves the currently open file with a new file name. When this command is clicked, the Save As file dialog box comes up.




To save the file with a new file name, locate the directory where you want to save the file, specify a file name for it, then click “OK”.  
Click “Cancel” to cancel.

## Edit Menu

 **New Question-Response:** This command creates a new question and response text box pair.

 **New Label:** This command creates a new label.

 **Delete:** This command deletes a label or a question-response pair. To delete a label, first click on the label, then click Edit: Delete. To delete a question-response pair, first click on a question box, then click Edit: Delete. You cannot delete a question box without deleting its corresponding response box. You cannot delete a response box directly; to delete a response box, you must delete its corresponding question box.

## View Menu

See Main Window: View Menu.

## **Window Menu**

**New Window:** This command opens a second window (view) of the same file.

**Cascade:** This command cascades all open windows.

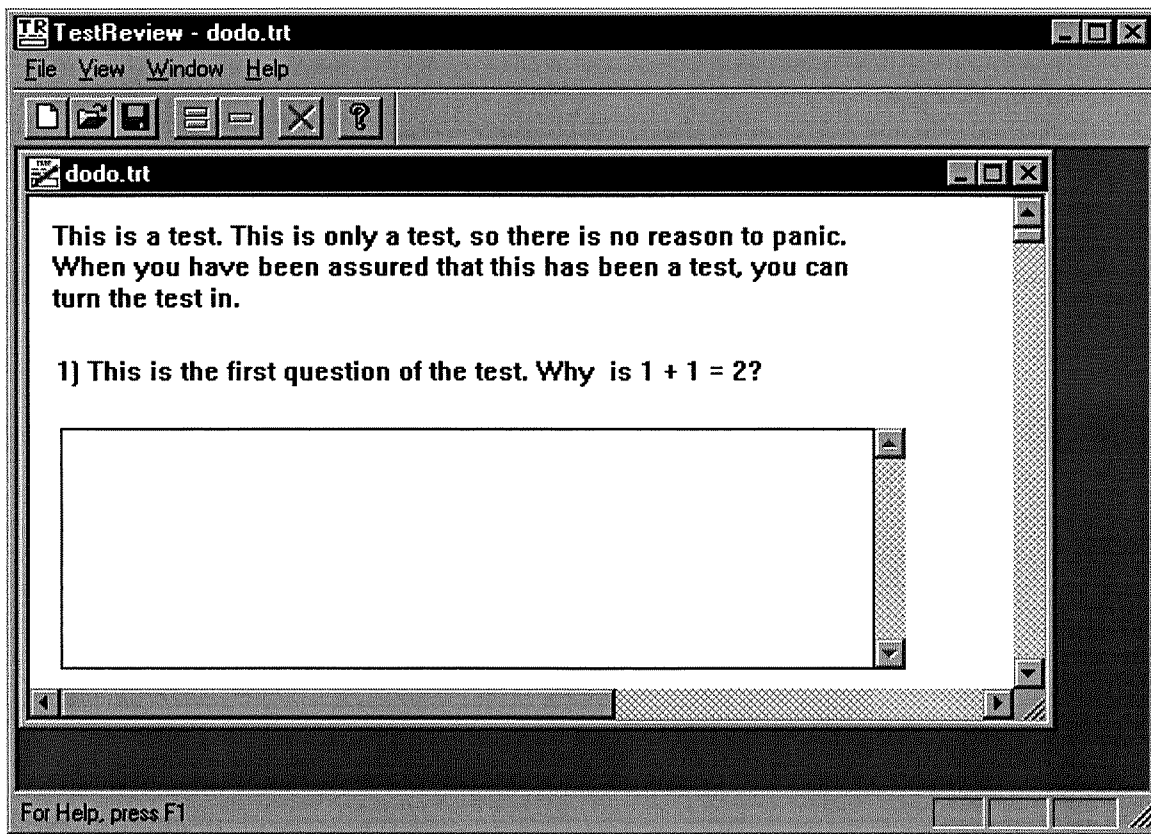
**Tile:** This command tiles all open windows.

**Arrange Icons:** This command automatically arranges all minimized icons.

## **Help Menu**

**See Main Window: View Menu.**

## Test View



### File Menu

New, Open, MRU List, Exit: Same as in Main Window.

Close, Save, Save As: Same as in Design View.

### View Menu

See Main Window: View Menu.

### Window Menu

See Design View: Window Menu.

## **Help Menu**

**See Main Window: View Menu.**

## Glossary

<b>Design View</b>	The view in which <b>instructors</b> create question (.TRQ) files. See also <b>Test View</b> .
<b>Instructor</b>	Someone who creates a question file in the Design View of TestReview.
<b>Label Box</b>	A text box which contains a <i>label</i> . A label is text other than a question that is displayed to the <b>student</b> . In the <b>Design View</b> , the <b>instructor</b> can type in a label box; in the <b>Test View</b> , a label box is read-only.
<b>Question Box</b>	A text box which contains the question. In the <b>Design View</b> , the <b>instructor</b> can type a question into a question box; in the <b>Test View</b> , a question box is read-only.
<b>Response Box</b>	A text box which contains the response. In the <b>Design View</b> , no text can be typed in a response box; in the <b>Test View</b> , a student can type text into a response box to answer a question.
<b>Student</b>	Someone who takes a test based on a question file created by an <b>instructor</b>
<b>Test View</b>	The view in which <b>students</b> take tests based on question files; The Test View generates .TRT files. See also <b>Design View</b> .

# ***TestReview<sup>TM</sup> 1.0***

## **Program Documentation**

---

## **Table of Contents**

<b>Introduction</b>	<b>3</b>
<b>Programming Environment</b>	<b>4</b>
<b>Overview of the TestReview Views</b>	<b>4</b>
<b>TestReview Class Hierarchy Chart</b>	<b>5</b>
<b>TestReview Classes</b>	<b>6</b>
<b>Classes Generated by the MFC Framework</b>	<b>6</b>
<b>CControlID</b>	<b>6</b>
<b>CDesignDoc</b>	<b>7</b>
<b>CDesignEdit</b>	<b>8</b>
<b>CDesignView</b>	<b>9</b>
<b>CResponseEdit</b>	<b>11</b>
<b>CStaticEdit</b>	<b>11</b>
<b>CTestDoc</b>	<b>12</b>
<b>CTestView</b>	<b>13</b>
<b>Supporting Structure Classes</b>	<b>14</b>
<b>Global Variables and Functions</b>	<b>14</b>



---

**Glossary** **15**

**Source Listing** **16**

## Introduction

TestReview is a program designed to let instructors create electronic tests that students can take on a computer. TestReview provides an interface that lets instructors visually create text boxes in which they can type questions or labels, and lets them create other text boxes in which students can respond to questions. This information can be saved to a question file. Students that wish to take the test recorded in a question file can create a test file, which is based on a question file created by an instructor.

This document is the program documentation for version 1.0 of TestReview. This version has the basic functionality required to fully enable instructors create electronic tests and let students take these tests. This document focuses on the inner workings of the TestReview program. To understand the implementation of TestReview, it is important to first read *TestReview 1.0 User's Manual* to understand TestReview's features. The rest of this documentation will assume that the programmer is familiar with how TestReview works and the terminology used in the *User's Manual*.

Included with this documentation is a 3.5-inch floppy disk with the TestReview executable (TestReview.exe) and TestReview source files. The files included are:

- ChildFrm.cpp; ChildFrm.h;
- DesignDoc.cpp; DesignDoc.h;
- DesignEdit.cpp; DesignEdit.h;
- DesignView.cpp; DesignView.h;
- Globals.cpp;
- MainFrm.cpp; MainFrm.h;
- ResponseEdit.cpp; ResponseEdit.h;
- StaticEdit.cpp; StaticEdit.h;
- StdAfx.cpp; StdAfx.h;
- TestDoc.cpp; TestDoc.h;
- TestReview.cpp; TestReview.h;
- TestReview.rc; TestReview.rc2;
- TestView.cpp; TestView.h.
- ControlID.h;
- EditStruct.h
- QREdit.h;
- resource.h

Other project files are:

- bmp00001.bmp; DesignDoc.ico; designnty.bmp; mainfram.bmp; TestDoc.ico;
- TestReview.ico; TestReview.reg; testtype.bmp

## Programming Environment

TestReview was created using Microsoft Visual C++ 4.0. Visual C++ is presented by Microsoft Developers' Studio (MSDEV), which is an integrated design environment for software development. Because a lot of the code in this program was automatically generated by Visual C++'s AppWizard, the code is best viewed using the MSDEV environment; if the source files are viewed as-is, they might appear somewhat untidy and disordered.

TestReview was compiled using the Microsoft Windows NT 4.0 operating system, and the code generated can run on all Win32 platforms (Windows 95, Windows NT and compatible operating systems).

A Compaq Proliant 1500 PC with a Pentium 166 MHz processor, 64 MB RAM and a 2 GB SCSI hard drive was used to create the program.

TestReview was created using Microsoft Foundation Classes (MFC) 4.0, and the MFC framework was used as much as possible. In this document, MFC features are only briefly described and it is assumed that the user is familiar with MFC or has access to MFC documentation. Thus, only those MFC classes, data members and functions that I modified or created from scratch are described here. For a description of any MFC structures or functions, please refer to Microsoft Foundation Classes 4.0 documentation.

## Overview of the TestReview Views

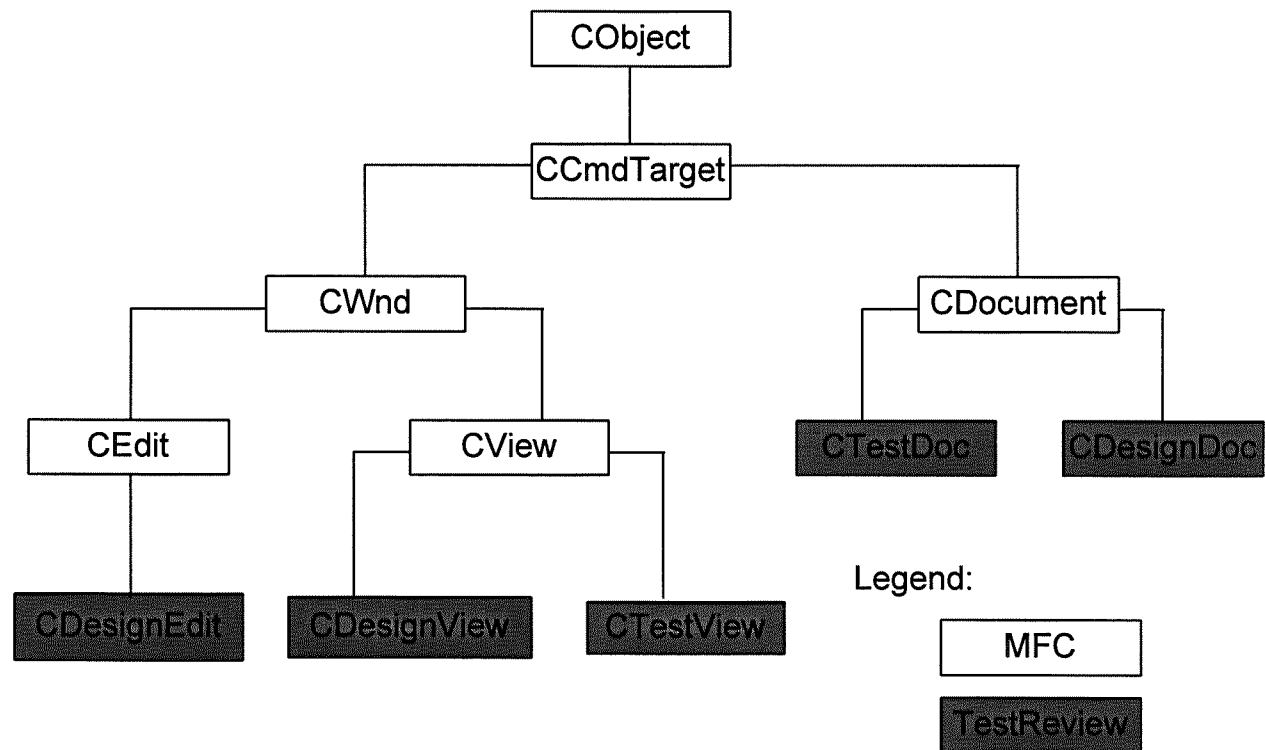
### The Design View

The Design View is the view that instructors use to create tests. In TestReview, the Design View is implemented with the CDesignDoc document class, the CDesignView view class, and CDesignEdit edit box class. These classes will be described under **TestReview Classes**.

### The Test View

The Test View is the view that allows a student to take a test from a question file. In TestReview, the Test View is implemented with the CTestDoc document class, the CTestView view class, and CStaticEdit and CResponseEdit edit box classes. These classes will be described under **TestReview Classes**.

# TestReview Class Hierarchy



## TestReview Classes

### Classes Generated by the MFC Framework

```
class CAboutDlg : public CDialog
    The About dialog box.

class CChildFrame : public CMDIChildWnd
    The child frame window for the MDI frame window. This child frame serves all views in
    TestReview

class CMainFrame : public CMDIFrameWnd
    The MDI frame window for TestReview.

class CTestReviewApp : public CWinApp
    The windows application class; there is a single instance of this class in TestReview:
    CTestReviewApp theApp.
```

### CControlID

CControlID stores a unique control ID and automatically increments this ID whenever it is assigned to a control. Its purpose is to maintain unique control IDs within any view which inherits it. CControlID is not based on MFC.

#### Data Members:

`m_id`: Stores the next ID number to be assigned.

#### Functions:

`CControlID()`  
The constructor initializes the control ID to 1.

`AssignID()`  
Returns a control ID number and increments the control ID.

## CDesignDoc

CDesignDoc stores the actual data needed by the Design View in TestReview. It holds all the data necessary to save and reload created tests.

### Base MFC Class: CDocument

The CDocument class provides the basic functionality for user-defined document classes. A document represents the unit of data that the user typically opens with the File Open command and saves with the File Save command.

### Data members:

`m_labelList`: Stores a list of CEditDatas containing essential label information.  
`m_QRList`: Stores a list of CQRDatas containing essential question-response pair information.

### Functions:

`Serialize()`  
Stores and loads document data to and from disk.

### Friends:

`class CDesignView`  
Needs to access `m_labelList` and `m_QRList` to initialize the view state.

`class CTestDoc`  
Needs to access `m_labelList` and `m_QRList` when using a DesignDoc as a template to create a TestDoc

`::UpdateDesignDoc( CDesignDoc*, CDesignView* )`  
Needs to access `m_labelList` and `m_QRList` to update the current document state.

## CDesignEdit

CDesignEdit implements the modifiable edit controls for the Design View in TestReview. In addition to standard edit control functionality, DesignEdit controls are moveable and resizable, and also display specialized behaviour, depending on whether they are questions, responses or labels.

### Base MFC Class: CEdit

The CEdit class provides the functionality of a Windows edit control. An edit control is a rectangular child window in which the user can enter text.

### Data members:

`m_tracker`: A rectangle tracker used to store the position of the DesignEdit control and also to track the control when it is being moved.  
`m_type`: The type of DesignEdit; one of `DE_QUESTION`, `DE_RESPONSE`, and `DE_LABEL`.  
`pView`: The view that contains the DesignEdit control.  
`pDoc`: Pointer to `pView`'s underlying document.

### Functions:

`CDesignEdit()`  
Initializes `m_tracker`'s style.

`Create()`  
Creates a new DesignEdit control.

`GetWindowRect()`  
Gets the window rectangle in client coordinates (ie., relative to the parent window).

`GetType()`  
Returns `m_type`.

`OnKillFocus()`  
Sets the DesignEdit control to read-only mode when it loses the input focus.

`OnLButtonDbClick()`  
For question- and label-type DesignEdit controls, turns off the read-only mode.

`OnLButtonDown()`  
Lets the user move the DesignEdit control to a new position.

OnSetCursor()

Changes the cursor shape when the cursor is over the DesignEdit control.

OnSize()

Updates the document when the DesignEdit control is resized.

OnUpdate()

Updates the document when the text is updated.

UpdateDoc()

Tells the document that changes have been made.

## CDesignView

CDesignView presents the user interface for the Design View in TestReview. It holds the DesignEdit controls that the user creates and modifies, and it acts as an interface between the underlying CDesignDoc and the user.

### Base MFC Class: CView

CView provides the basic functionality for user-defined view classes. A view is attached to a document and acts as an intermediary between the document and the user: the view renders an image of the document on the screen or printer and interprets user input as operations upon the document.

### Data members:

m\_labelList: Stores the list of label CDesignEdits.

m\_QRList: Stores the list of question-response pair CQREdits.



**Functions:**

CDesignView()

Constructor. Initializes m\_QRList and m\_labelList to empty lists.

GetDocument()

Returns a pointer to CDesignDoc underlying the CDesignView.

OnEditNewQR()

Responds to the “New Question-Response” command by creating a new question-response pair and adding them to m\_QRList.

OnEditNewLabel()

Responds to the “New Label” command by creating a new label and adding it to m\_labelList.

OnEditDelete()

Responds to the “Delete” command by deleting a DesignEdit.

OnInitialUpdate()

Initializes the view and loads any stored questions, responses and labels.

OnLButtonDown()

On left mouse click, set the currently selected CDesignEdit to read only mode.

**Friends:**

::UpdateDesignDoc( CDesignDoc\*, CDesignView\* )

Needs to access m\_labelList and m\_QRList to update the current document state.

::DeleteDesignEdit( CDesignEdit\*, CView\* )

Needs to access m\_labelList and m\_QRList to delete DesignEdits from them.

## CResponseEdit

CResponseEdit is an edit control that notifies its view's underlying document when it is modified.

### Base MFC Class: CEdit

The CEdit class provides the functionality of a Windows edit control. An edit control is a rectangular child window in which the user can enter text.

### Functions:

OnChange()  
Notify the document that this edit control has changed

## CStaticEdit

CStaticEdit is an edit control that does not receive focus and does not change the cursor. Its behaviour is similar to that of a static control, but its default color is that of an edit control.

### Base MFC Class: CEdit

The CEdit class provides the functionality of a Windows edit control. An edit control is a rectangular child window in which the user can enter text.

### Functions:

OnChar()  
Does not accept any characters from the keyboard.

OnRButtonDown()  
Does nothing when the right mouse button is clicked.

OnSetCursor()  
Lets the view handle the cursor shape.

OnSetFocus()  
Does not accept input focus.

## CTestDoc

CTestDoc stores the actual data needed by the Test View in TestReview. It holds all the data necessary to save and reload a test template and the responses given.

### Base MFC Class: CDocument

The CDocument class provides the basic functionality for user-defined document classes. A document represents the unit of data that the user typically opens with the File Open command and saves with the File Save command.

### Data members:

`m_labelList`: Stores a list of `CEditDatas` containing essential label information.  
`m_questionList`: Stores a list of `CEditDatas` containing essential question box information.  
`m_responseList`: Stores a list of `CEditDatas` containing essential response box information.

### Functions:

`OnNewDocument()`  
Loads a `DesignDoc` template based on an existing `DesignDoc` file, and creates a new `TestDoc`.  
  
`Serialize()`  
Stores and loads document data to and from disk.

### Friends:

`class CTestView`  
Needs to access `m_labelList` and `m_QRList` to initialize the view state.  
  
`::UpdateTestDoc( CTestDoc*, CTestView* )`  
Needs to access `m_responseList` to update the current document state.

## CTestView

CTestView represents the user interface for the Test View in TestReview. It holds the edit box controls that represent labels, questions and response text boxes; CTestView acts as the interface between the underlying CTestDoc and the user.

### Base MFC Class: CView

CView provides the basic functionality for user-defined view classes. A view is attached to a document and acts as an intermediary between the document and the user: the view renders an image of the document on the screen or printer and interprets user input as operations upon the document.

### Data members:

m\_labelList: Stores a list of label StaticEdits.  
m\_questionList: Stores a list of question StaticEdits.  
m\_responseList: Stores a list of response ResponseEdits.

### Functions:

OnInitialUpdate()  
Initializes the view and loads any stored questions, responses and labels.

### Friends:

::UpdateTestDoc( CTestDoc\*, CTestView\* )  
Needs to access m\_responseList to update the current document state.

## Supporting Structure Classes

CEditData, CQRData, and CQREdit are simple classes which store document data or DesignEdit controls.

### CEditData

The CEditData class stores the text stored by a DesignEdit control and rect, the rectangle position of the control. It is used by CDesignDoc, CTestDoc, and by the global functions.

### CQRData

The CQRData class stores two CEditData structures representing a question-response pair. They are question and response. It is used by CDesignDoc, CTestDoc, and by the global functions.

### CQREdit

The CQREdit class contains an actual question-response pair. Its members are question and response. It is used by CDesignView.

## Global Variables and Functions

### Variables:

TestReviewApp theApp  
The TestReview application object.

### Functions:

void DeleteDesignEdit( CDesignEdit\*, CView\* )  
Deletes a DesignEdit from its containing DesignView

void UpdateDesignDoc( CDesignDoc\*, CDesignView\* )  
Updates a DesignDoc with the current state of a DesignView

void UpdateTestDoc( CTestDoc\*, CTestView\* )  
Updates a TestDoc with the current state of a TestView

## Glossary

<b>Design View</b>	The view in which <b>instructors</b> create question (.TRQ) files. See also <b>Test View</b> .
<b>Instructor</b>	Someone who creates a question file in the Design View of TestReview.
<b>Label Box</b>	A text box which contains a <i>label</i> . A label is text other than a question that is displayed to the <b>student</b> . In the <b>Design View</b> , the <b>instructor</b> can type in a label box; in the <b>Test View</b> , a label box is read-only.
<b>MFC</b>	Microsoft Foundation Classes. Microsoft's proprietary class hierarchy that supports Windows programming with C++.
<b>Question Box</b>	A text box which contains the question. In the <b>Design View</b> , the <b>instructor</b> can type a question into a question box; in the <b>Test View</b> , a question box is read-only.
<b>Response Box</b>	A text box which contains the response. In the <b>Design View</b> , no text can be typed in a response box; in the <b>Test View</b> , a student can type text into a response box to answer a question.
<b>Student</b>	Someone who takes a test based on a question file created by an <b>instructor</b>
<b>Test View</b>	The view in which <b>students</b> take tests based on question files; The Test View generates .TRT files. See also <b>Design View</b> .

## TestReview Source Listing

Source files are listed alphabetically.

### CHILDFRM.CPP;

```
// ChildFrm.cpp : implementation of the CChildFrame class
//

#include "stdafx.h"
#include "TestReview.h"

#include "ChildFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CChildFrame

IMPLEMENT_DYNCREATE(CChildFrame, CMDIChildWnd)

BEGIN_MESSAGE_MAP(CChildFrame, CMDIChildWnd)
    //{{AFX_MSG_MAP(CChildFrame)
    // NOTE - the ClassWizard will add and remove mapping macros here.
    //      DO NOT EDIT what you see in these blocks of generated code !
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CChildFrame construction/destruction

CChildFrame::CChildFrame()
{
    // TODO: add member initialization code here
}

CChildFrame::~CChildFrame()
{
}

////////////////////////////////////
// CChildFrame diagnostics

#ifdef _DEBUG
void CChildFrame::AssertValid() const
{

```

```

        CMDIChildWnd::AssertValid();
    }

void CChildFrame::Dump(CDumpContext& dc) const
{
    CMDIChildWnd::Dump(dc);
}

#endif // _DEBUG

////////////////////////////////////
// CChildFrame message handlers

```

## CHILDfrm.H;

```

// ChildFrm.h : interface of the CChildFrame class
//
////////////////////////////////////

class CChildFrame : public CMDIChildWnd
{
    DECLARE_DYNCREATE(CChildFrame)
public:
    CChildFrame();

    // Attributes
public:

    // Operations
public:

    // Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CChildFrame)
    //}}AFX_VIRTUAL

    // Implementation
public:
    virtual ~CChildFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

    // Generated message map functions
protected:
    //{{AFX_MSG(CChildFrame)
    // NOTE - the ClassWizard will add and remove member functions
here.
        //      DO NOT EDIT what you see in these blocks of generated code!
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

```



```
////////////////////////////////////
```

## CONTROLID.H;

```
// ControlID.h: header file
//

#ifndef CONTROLID_H
#define CONTROLID_H

// CControlID stores a unique control id and automatically increments it
// whenever it is assigned
class CControlID {
private:
    UINT m_id; // the control id

public:
    CControlID() { m_id = 1; }

protected:
    // assign and increment the control id
    UINT AssignID() { return m_id++; }
};

#endif // #ifndef CONTROLID_H
```

## DESIGNDOC.CPP;

```
// DesignDoc.cpp : implementation of the CDesignDoc class
//

#include "stdafx.h"
#include "TestReview.h"

#include "DesignDoc.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CDesignDoc

IMPLEMENT_DYNCREATE(CDesignDoc, COleDocument)

BEGIN_MESSAGE_MAP(CDesignDoc, COleDocument)
```

[illegible]

```

void CDesignDoc::Serialize(CArchive& ar)
{
    class CDesignEdit;
    class CDesignView;
    void UpdateDesignDoc( CDesignDoc* pDoc, CDesignView* pView );

    if (ar.IsStoring())
    // Store data: first update the document data, then store it.
    {
        POSITION pos;
        UpdateDesignDoc( this, (CDesignView*) GetNextView(
pos=GetFirstViewPosition() ) );

        ar << m_QRList.GetCount();    // number of QRDatas in m_QRList

        // store each QRData from the list
        pos = m_QRList.GetHeadPosition();
        CQRData* pQRData;
        while ( pos ) {
            pQRData = (CQRData*) m_QRList.GetNext( pos );
            ar << pQRData->question.text << pQRData->question.rect;
            ar << pQRData->response.text << pQRData->response.rect;
        }

        ar << m_labellList.GetCount(); // number of EditDatas in
m_labellList

        // store each label from the list
        pos = m_labellList.GetHeadPosition();
        CEditData* pLabel;
        while ( pos ) {
            pLabel = (CEditData*) m_labellList.GetNext( pos );
            ar << pLabel->text << pLabel->rect;
        }

    }
    else
    // Retrieve data. Note that data is retrieved only when a new document
    // is created, meaning that the list data members will necessarily be
    // empty.
    {

        int list_length; // number of list items to retrieve

        m_QRList.RemoveAll(); // sets list to NULL
        ar >> list_length;    // number of QRDatas to retrieve
        for ( int i=0; i < list_length; i++ ) {
            CQRData* pQRData = new CQRData;
            ar >> pQRData->question.text >> pQRData->question.rect;
            ar >> pQRData->response.text >> pQRData->response.rect;

            m_QRList.AddTail( pQRData );
        }

        m_labellList.RemoveAll(); // sets list to NULL
        ar >> list_length;    // number of labels to retrieve
        for ( i=0; i < list_length; i++ ) {
            CEditData* pLabel = new CEditData;
            ar >> pLabel->text >> pLabel->rect;
        }
    }
}

```

```

        m_labelList.AddTail( pLabel );
    }

    // Calling the base class COleDocument enables serialization
    // of the container document's COleClientItem objects.
    COleDocument::Serialize(ar);
}

////////////////////////////////////
// CDesignDoc diagnostics

#ifdef _DEBUG
void CDesignDoc::AssertValid() const
{
    COleDocument::AssertValid();
}

void CDesignDoc::Dump(CDumpContext& dc) const
{
    COleDocument::Dump(dc);
}
#endif // _DEBUG

////////////////////////////////////
// CDesignDoc commands

```

## DESIGNDOC.H;

```

// DesignDoc.h : interface of the CDesignDoc class
//
////////////////////////////////////

#ifndef DESIGNDOC_H
#define DESIGNDOC_H

#include "DesignEdit.h"
#include "EditStruct.h"

class CDesignDoc : public COleDocument
{
    friend class CDesignView;
    friend class CTestDoc;
    friend void UpdateDesignDoc( CDesignDoc*, CDesignView* );

protected: // create from serialization only
    CPtrList    m_QRList;        // lists of data for question-response
pairs
    CPtrList    m_labelList;    // and labels respectively
    CDesignDoc();
    DECLARE_DYNCREATE(CDesignDoc)

```

```

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CDesignDoc)
    public:
        virtual BOOL OnNewDocument();
        virtual void Serialize(CArchive& ar);
    //}AFX_VIRTUAL

// Implementation
public:
    virtual ~CDesignDoc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
    //{AFX_MSG(CDesignDoc)
    // NOTE - the ClassWizard will add and remove member functions
    here.
        // DO NOT EDIT what you see in these blocks of generated code !
    //}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

#endif // #ifdef DESIGNDOC_H

```

## DESIGNEDIT.CPP;

```

// DesignEdit.cpp : implementation file
//

#include "stdafx.h"
#include "TestReview.h"
#include "DesignEdit.h"
#include "EditStruct.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

```

```

IMPLEMENT_DYNAMIC( CDesignEdit, CEdit )    // gives access to runtime
information

////////////////////////////////////
// CDesignEdit

CDesignEdit::CDesignEdit()
{
    // initialize members
    m_tracker.m_nStyle = CRectTracker::dottedLine;
}

CDesignEdit::~CDesignEdit()
{
}

BEGIN_MESSAGE_MAP(CDesignEdit, CEdit)
    //{AFX_MSG_MAP(CDesignEdit)
    ON_WM_KILLFOCUS()
    ON_WM_LBUTTONDOWNCLK()
    ON_WM_SIZE()
    ON_WM_LBUTTONDOWN()
    ON_WM_SETCURSOR()
    ON_CONTROL_REFLECT(EN_UPDATE, OnUpdate)
    //}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CDesignEdit message handlers

void CDesignEdit::OnKillFocus(CWnd* pNewWnd)
// Set to read-only mode when the input focus is lost
{
    CEdit::OnKillFocus(pNewWnd);

    SetReadOnly();
}

void CDesignEdit::OnLButtonDblClk(UINT nFlags, CPoint point)
{
    switch ( m_type ) {
    case DE_QUESTION:
    case DE_LABEL:           // turn off read-only state
        SetReadOnly( FALSE );
        SetFocus();
        CEdit::OnLButtonDblClk(nFlags, point);
        break;
    case DE_RESPONSE: // do nothing
        SetFocus();
        break;
    }
}

void CDesignEdit::OnSize(UINT nType, int cx, int cy)
{
    CEdit::OnSize(nType, cx, cy);
}

```

```

        // update the stored window rectangle coordinates
        GetWindowRect( m_tracker.m_rect, pView );

        UpdateDoc();
    }

void CDesignEdit::GetWindowRect( LPRECT lpRect, CView *pView ) const
// gets the window rectangle in client coordinates (ie., relative to the
parent view)
{
    CEdit::GetWindowRect( lpRect );    // gives screen coordinates

    CRect rViewRect; // the view's bounding rectangle
    pView->GetWindowRect( &rViewRect );
    lpRect->left -= rViewRect.left;
    lpRect->right -= rViewRect.left;
    lpRect->top -= rViewRect.top;
    lpRect->bottom -= rViewRect.top;
}

void CDesignEdit::OnLButtonDown(UINT nFlags, CPoint point)
// if the box is in read-only mode, move it
{
    SetFocus();

    if ( GetStyle() & ES_READONLY ) {

        // set the tracker
        GetWindowRect( (LPRECT) &(m_tracker.m_rect), pView );

        // adjust point relative to tracker
        point.x += m_tracker.m_rect.left;
        point.y += m_tracker.m_rect.top;

        if ( m_tracker.Track( pView, point ) ) {
            // successfully moved
            this->MoveWindow( m_tracker.m_rect );
            BringWindowToTop();
            pView->Invalidate();

            UpdateDoc();
        }

        return;
    }

    CEdit::OnLButtonDown(nFlags, point);    // executed if box is not
read-only
}

BOOL CDesignEdit::Create( DWORD dwStyle, const RECT& rect, CWnd* pParentWnd,
UINT nID,
                                DETYPE teType, CString sText )    // default
sText = ""
{
    // set the view and document pointers

```

```

    pView = (CView *) pParentWnd; // note that CDesignEdit only lets views
be its parents
    pDoc = pView->GetDocument();

    // set the window style according to the type of DesignEdit
    dwStyle = dwStyle | WS_THICKFRAME | WS_BORDER | WS_CHILD | WS_VISIBLE
               | WS_TABSTOP | ES_READONLY;
    switch( m_type = teType ) {
    case DE_QUESTION:
    case DE_LABEL:
        dwStyle = dwStyle | ES_MULTILINE;
        break;
    case DE_RESPONSE:
        dwStyle = dwStyle | WS_VSCROLL;
        break;
    }

    BOOL bRet = CEdit::Create( dwStyle, rect, pParentWnd, nID );

    SetWindowText( sText );

    return bRet;
}

BOOL CDesignEdit::OnSetCursor(CWnd* pWnd, UINT nHitTest, UINT message)
// change the cursor when it is over the edit box
{
    if ( ( GetStyle() & ES_READONLY )
        && ( nHitTest != HTTOPLEFT )
        && ( nHitTest != HTTOP )
        && ( nHitTest != HTTOPRIGHT )
        && ( nHitTest != HTLEFT )
        && ( nHitTest != HTBOTTOMLEFT )
        && ( nHitTest != HTBOTTOM )
        && ( nHitTest != HTBOTTOMRIGHT )
        && ( nHitTest != HTRIGHT ) ) {
        // DesignEdit is read-only and not on a border, so change the
cursor
        SetCursor( LoadCursor( NULL, IDC_SIZEALL ) );
        return TRUE;
    }

    return CEdit::OnSetCursor(pWnd, nHitTest, message);
}

inline void CDesignEdit::UpdateDoc()
{
    pDoc->SetModifiedFlag();
}

void CDesignEdit::OnUpdate()
{
    UpdateDoc();
}

```



## DESIGNEDIT.H;

```
// DesignEdit.h : header file
//

/////////////////////////////////////////////////////////////////
// CDesignEdit window

#ifndef DesignEdit_H
#define DesignEdit_H

class CDesignEdit : public CEdit
{
    DECLARE_DYNAMIC( CDesignEdit )      // gives access to runtime
    information

    // Construction
public:
    CDesignEdit();

    // Data Members, etc.
public:
    // DETYPE specifies the possible types of CDesignEdit
    typedef enum { DE_QUESTION, DE_RESPONSE, DE_LABEL } DETYPE;
    CRectTracker m_tracker; // The tracking rectangle for moving the
    DesignEdit.

                                // In addition,
    m_tracker.m_rect stores the                                // current client rectangle
    position of the                                            // DesignEdit.

protected:
    void UpdateDoc();
    DETYPE m_type;                                // the type of the current CDesignEdit

    CView* pView;                                // pointer to the parent view
    CDocument* pDoc;                            // pointer to the parent view's document

//private:
//    static UINT m_id; // the automatic ID for the next DesignEdit control
//    created

    // Overriden Member Functions
public:
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CDesignEdit)
    //{AFX_VIRTUAL

    void GetWindowRect( LPRECT lpRect, CView *pView ) const;

    BOOL Create( DWORD dwStyle,                                // window style
```



```

IMPLEMENT_DYNCREATE(CDesignView, CScrollView)

BEGIN_MESSAGE_MAP(CDesignView, CScrollView)
    //{AFX_MSG_MAP(CDesignView)
    ON_WM_LBUTTONDOWN()
    ON_COMMAND(ID_EDIT_NEWQR, OnEditNewQR)
    ON_COMMAND(ID_EDIT_NEWLABEL, OnEditNewLabel)
    ON_COMMAND(ID_EDIT_DELETE, OnEditDelete)
    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CScrollView::OnFilePrintPreview)
END_MESSAGE_MAP()

////////////////////////////////////
// CDesignView construction/destruction

CDesignView::CDesignView()
{
    // initialize the data members

    m_QRList.RemoveAll();
    m_labelList.RemoveAll();
}

CDesignView::~CDesignView()
{
    // free the view data
    CQREdit* pQREdit;
    while ( !m_QRList.IsEmpty() ) {
        pQREdit = (CQREdit*) m_QRList.RemoveHead();
        delete pQREdit;
    }
    m_QRList.RemoveAll();

    CDesignEdit* pDesignEdit;
    while ( !m_labelList.IsEmpty() ) {
        pDesignEdit = (CDesignEdit*) m_labelList.RemoveHead();
        delete pDesignEdit;
    }
    m_labelList.RemoveAll();
}

////////////////////////////////////
// CDesignView drawing

void CDesignView::OnDraw(CDC* pDC)
{
    CDesignDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
}

void CDesignView::OnInitialUpdate()
// initialize the view, including loading any stored DesignEdit lists

```

```

{
    CScrollView::OnInitialUpdate();

    CSize sizeTotal;
    // calculate the total size of this view
    sizeTotal.cx = 850;
    sizeTotal.cy = 10000;
    SetScrollSizes(MM_TEXT, sizeTotal);

    CDesignDoc* pDoc = GetDocument();

    // Construct any existing questions and responses from the document
data.
    CQREdit* pQREdit;
    CQRData* pQRData;
    for ( POSITION pos = pDoc->m_QRList.GetHeadPosition(); pos != NULL; ) {
        pQRData = (CQRData*) pDoc->m_QRList.GetNext( pos );

        this->m_QRList.AddTail( pQREdit = new CQREdit );

        pQREdit->question.Create( NULL, pQRData->question.rect, this,
AssignID(),

        CDesignEdit::DE_QUESTION, pQRData->question.text );
        pQREdit->response.Create( NULL, pQRData->response.rect, this,
AssignID(),

        CDesignEdit::DE_RESPONSE, pQRData->response.text );
    }

    // Construct any existing labels from the document data.
    CDesignEdit* pDesignEdit;
    CEditData* pEditData;
    for ( pos = pDoc->m_labelList.GetHeadPosition(); pos != NULL; ) {
        pEditData = (CEditData*) pDoc->m_labelList.GetNext( pos );

        this->m_labelList.AddTail( pDesignEdit = new CDesignEdit );

        pDesignEdit->Create( NULL, pEditData->rect, this, AssignID(),
                                CDesignEdit::DE_LABEL, pEditData-
>text );
    }

    // don't set off a false alarm
    pDoc->SetModifiedFlag( FALSE );
}

////////////////////////////////////
// CDesignView diagnostics

#ifdef _DEBUG
void CDesignView::AssertValid() const
{
    CScrollView::AssertValid();
}

void CDesignView::Dump(CDumpContext& dc) const
{

```

```

        CScrollView::Dump(dc);
    }

    CDesignDoc* CDesignView::GetDocument() // non-debug version is inline
    {
        ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CDesignDoc)));
        return (CDesignDoc*)m_pDocument;
    }
#endif // _DEBUG

////////////////////////////////////
// CDesignView message handlers
//

void CDesignView::OnLButtonDown(UINT nFlags, CPoint point)
// on left mouse click, set edit box to read only mode
{
    ((CDesignEdit*) GetFocus())->SetReadOnly();

    CScrollView::OnLButtonDown(nFlags, point);
}

void CDesignView::OnEditNewQR()
// add a new question-response pair
{
    CQREdit* pQR = new CQREdit;

    pQR->question.Create( NULL, CRect(10,10,300,100), this, AssignID(),
                                CDesignEdit::DE_QUESTION );
    pQR->response.Create( NULL, CRect(10,150,300,250), this, AssignID(),
                                CDesignEdit::DE_RESPONSE );

    m_QRList.AddTail( pQR );
}

void CDesignView::OnEditNewLabel()
// add a new label
{
    CDesignEdit* pEdit = new CDesignEdit;

    pEdit->Create( NULL,
position          CRect(10,10,300,100), // arbitrary initial
                                this, AssignID(), CDesignEdit::DE_LABEL );

    m_labelList.AddTail( pEdit );
}

void CDesignView::OnEditDelete()
// deletes a DesignEdit
{
    // get the runtime class of the current window with the input focus
    CRuntimeClass* pRT = GetFocus()->GetRuntimeClass();
    if ( strcmp( pRT->m_lpszClassName, "CDesignEdit" ) == 0 )
        // the current object is a DesignEdit
        DeleteDesignEdit( (CDesignEdit*) GetFocus(), this );
}

```

```

        return;
    }

```

## DESIGNVIEW.H;

```

// DesignView.h : interface of the CDesignView class
//
/////////////////////////////////////////////////////////////////

#ifndef DESIGNVIEW_H
#define DESIGNVIEW_H

#include "DesignEdit.h"
#include "ControlID.h"

class CDesignDoc;

class CDesignView : public CScrollView, protected CControlID
{
    friend void UpdateDesignDoc( CDesignDoc*, CDesignView* );
    friend void DeleteDesignEdit( CDesignEdit* pEdit, CView* pView );

protected: // create from serialization only
    CDesignView();
    DECLARE_DYNCREATE(CDesignView)

protected:
    CPtrList    m_labelList;        // list of all labels
    CPtrList    m_QRList;           // list of all QREdits

// Attributes
public:
    //BOOL OverlappedRect ( CRect& rect ) const;
    CDesignDoc* GetDocument();

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CDesignView)
    public:
        virtual void OnDraw(CDC* pDC); // overridden to draw this view
    protected:
        virtual void OnInitialUpdate(); // called first time after construct
    }AFX_VIRTUAL

// Implementation
public:
    virtual ~CDesignView();
#ifdef _DEBUG
    virtual void AssertValid() const;

```

```

        virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
   //{{AFX_MSG(CDesignView)
    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
    afx_msg void OnEditNewQR();
    afx_msg void OnEditNewLabel();
    afx_msg void OnEditDelete();
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

#ifdef _DEBUG // debug version in DesignView.cpp
inline CDesignDoc* CDesignView::GetDocument()
{ return (CDesignDoc*)m_pDocument; }
#endif

#endif // #ifndef DESIGNVIEW_H
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

## EDITSTRUCT.H

```

// EditStruct.h: header file
//
// This header declares CEditData, CQRData, and CQREdit, which are all
// supporting classes for the storage of Edit control data
//

#ifndef EDITSTRUCT_H
#define EDITSTRUCT_H

// CEditData stores the text and the rectangle position (in client
// coordinates) of a window control.
class CEditData {
public:
    CString      text; // the window text
    CRect rect; // rectangular position in client coordinates
};

//CQRData stores the EditData for a question- and a response-type DesignEdit.
class CQRData {
public:
    CEditData    question;
    CEditData    response;
};

#endif // #ifndef EDITSTRUCT_H

```

## GLOBALS.CPP;

```

////////////////////////////////////
// Globals.cpp.
// This file stores all global variables and functions, except
// TestReviewApp theApp, which is declared in TestReview.cpp

#include "stdafx.h"
#include "DesignView.h"
#include "DesignDoc.h"
#include "DesignEdit.h"
#include "EditStruct.h"
#include "QREdit.h"
#include "TestDoc.h"
#include "TestView.h"

////////////////////////////////////
// Global Variables

////////////////////////////////////
// Global Functions

void UpdateDesignDoc( CDesignDoc* pDoc, CDesignView* pView )
// updates the document with the current state of the view
{
    POSITION pos;

    // free the document data
    CQRData* pQRData;
    while ( !pDoc->m_QRList.IsEmpty() ) {
        pQRData = (CQRData*) pDoc->m_QRList.RemoveHead();
        delete pQRData;
    }
    pDoc->m_QRList.RemoveAll();

    CEditData* pEditData;
    while ( !pDoc->m_labelList.IsEmpty() ) {
        pEditData = (CEditData*) pDoc->m_labelList.RemoveHead();
        delete pEditData;
    }
    pDoc->m_labelList.RemoveAll();

    // Re-create the updated data:

    // save the scroll position
    CPoint saved_pt = pView->GetScrollPosition();

    // set the view to the zero position for accurate saving
    pView->ScrollToPosition( CPoint(0, 0) );

    // iterate through pView->m_QRList
    CQREdit* pQREdit;
    for ( pos = pView->m_QRList.GetHeadPosition(); pos != NULL; ) {

```



```

        pQREdit = (CQREdit*) pView->m_QRList.GetNext( pos );
        pDoc->m_QRList.AddTail( pQRData = new CQRData );
        // create a new QRData
        pQREdit->question.GetWindowText( pQRData->question.text );
        pQREdit->question.GetWindowRect( pQRData->question.rect, pView );

        pQREdit->response.GetWindowText( pQRData->response.text );
        pQREdit->response.GetWindowRect( pQRData->response.rect, pView );
    }

    // iterate through pView->m_labelList
    CDesignEdit* pLabel;
    for ( pos = pView->m_labelList.GetHeadPosition(); pos != NULL; ) {
        pLabel = (CDesignEdit*) pView->m_labelList.GetNext( pos );
        pDoc->m_labelList.AddTail( pEditData = new CEditData );
        // create a new EditData
        pLabel->GetWindowText( pEditData->text );
        pLabel->GetWindowRect( pEditData->rect, pView );
    }

    // restore the position of the view
    pView->ScrollToPosition( saved_pt );

    pDoc->SetModifiedFlag();
}

```

```

void UpdateTestDoc( CTestDoc* pDoc, CTestView* pView )
// updates the document with the current state of the view
{
    // iterate through pView->m_responseList and pDoc->m_QRList
    CEdit* pEdit; // traverses the view
    CQRData* pQRData; // traverses the doc
    POSITION viewPos, docPos;

    for ( viewPos = pView->m_responseList.GetHeadPosition(),
          docPos = pDoc->m_QRList.GetHeadPosition();
          viewPos != NULL; ) {
        pEdit = (CEdit*) pView->m_responseList.GetNext( viewPos );
        pQRData = (CQRData*) pDoc->m_QRList.GetNext( docPos );

        pEdit->GetWindowText( pQRData->response.text );
    }

    pDoc->SetModifiedFlag();
}

```

```

void DeleteDesignEdit( CDesignEdit* pEdit, CView* pGenericView )
// deletes the DesignEdit from its containing DesignView
{
    CDesignView* pView = (CDesignView*) pGenericView;
    POSITION pos;
    CString message;
    CQREdit* pQR; // pointer to a question-edit pair

    switch ( pEdit->GetType() )
    {
        case CDesignEdit::DE_RESPONSE:

```

```

        message = "You cannot delete a response box directly; ";
        message += "to delete a response box, you need to delete its ";
        message += "corresponding question box.";
        AfxMessageBox( message );

        return;
        break;

    case CDesignEdit::DE_QUESTION:
        message = "Are you sure you want to delete a question box? ";
        message += "Its corresponding response box will also be
deleted.";
        if ( AfxMessageBox( message, MB_OKCANCEL ) == IDCANCEL )
            return;

        // Use the question to refer to the position of the entire
        // question-response pair; note that this depends on the fact that
        // the first item in a QR pair is the question.
        {
            pos = pView->m_QRList.Find( (CQREdit*) pEdit );
            pQR = (CQREdit*) pView->m_QRList.GetAt( pos );
            pView->m_QRList.RemoveAt( pos );
            delete pQR;
        }

        break;

    case CDesignEdit::DE_LABEL:
        message = "Are you sure you want to delete a label box?";
        if ( AfxMessageBox( message, MB_YESNO ) == IDNO )
            return;

        // delete the label
        pos = pView->m_labelList.Find( pEdit );
        pView->m_labelList.RemoveAt( pos );
        delete pEdit;

        break;
    }

    return;
}

```

## MAINFRM.CPP;

```

// MainFrm.cpp : implementation of the CMainFrame class
//

#include "stdafx.h"
#include "TestReview.h"

#include "MainFrm.h"

```

```

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMainFrame

IMPLEMENT_DYNAMIC(CMainFrame, CMDIFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CMDIFrameWnd)
    //{{AFX_MSG_MAP(CMainFrame)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        //      DO NOT EDIT what you see in these blocks of generated code !
    ON_WM_CREATE()
    //}}AFX_MSG_MAP
    // Global help commands
    ON_COMMAND(ID_HELP_FINDER, CMDIFrameWnd::OnHelpFinder)
    ON_COMMAND(ID_HELP, CMDIFrameWnd::OnHelp)
    ON_COMMAND(ID_CONTEXT_HELP, CMDIFrameWnd::OnContextHelp)
    ON_COMMAND(ID_DEFAULT_HELP, CMDIFrameWnd::OnHelpFinder)
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR,           // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here
}

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CMDIFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBar.Create(this) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;          // fail to create
    }

    if (!m_wndStatusBar.Create(this) ||

```

```

        !m_wndStatusBar.SetIndicators(indicators,
            sizeof(indicators)/sizeof(UINT))
    {
        TRACE0("Failed to create status bar\n");
        return -1;        // fail to create
    }

    // TODO: Remove this if you don't want tool tips or a resizable toolbar
    m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
        CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);

    // TODO: Delete these three lines if you don't want the toolbar to
    // be dockable
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

    return 0;
}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CMDIFrameWnd::PreCreateWindow(cs);
}

////////////////////////////////////
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CMDIFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CMDIFrameWnd::Dump(dc);
}

#endif // _DEBUG

////////////////////////////////////
// CMainFrame message handlers

```

## MAINFRM.H;

```

// MainFrm.h : interface of the CMainFrame class
//
////////////////////////////////////

```

```

#ifndef MAINFRAME_H
#define MAINFRAME_H

class CMainFrame : public CMDIFrameWnd
{
    DECLARE_DYNAMIC(CMainFrame)
public:
    CMainFrame();

    // Attributes
public:

    // Operations
public:

    // Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CMainFrame)
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    //}AFX_VIRTUAL

    // Implementation
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // control bar embedded members
    CStatusBar m_wndStatusBar;
    CToolBar m_wndToolBar;

    // Generated message map functions
protected:
    //{AFX_MSG(CMainFrame)
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    // NOTE - the ClassWizard will add and remove member functions
here.
    // DO NOT EDIT what you see in these blocks of generated code!
    //}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#endif // #ifndef MAINFRAME_H

```

## QREDIT.H;

```

// QREdit.h: header file
//
// CQREdit is a basic CObject-derived class that stores a question-type
// CDesignEdit object and a response-type CDesignEdit object.

```

```
//      It is used to enable CObject-derived functions such as Serialize().

#ifndef QREDIT_H
#define QREDIT_H

#include "DesignEdit.h"

class CQREdit {
public:
    CDesignEdit question;
    CDesignEdit response;
};

#endif      //      QREDIT_H
```

## RESOURCE.H

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by TestReview.rc
//
#define IDR_DESIGNTYPE_CNTR_IP          6
#define IDD_ABOUTBOX                    100
#define IDP_OLE_INIT_FAILED              100
#define IDP_FAILED_TO_CREATE             102
#define IDR_MAINFRAME                    128
#define IDR_DESIGNTYPE                  129
#define IDR_MAINFRAME3                   129
#define IDR_TESTTYPE                     130
#define IDR_DESIGNTYPE_CNTR_IP2          130
#define IDR_MAINFRAME1                   131
#define IDR_TESTTYPE1                    131
#define IDR_MAINFRAME2                   132
#define IDR_DESIGNTYPE_CNTR_IP1          133
#define IDR_DESIGNTYPE1                  134
#define ID_CANCEL_EDIT_CNTR              32768
#define ID_EDIT_NEWQR                    32771
#define ID_EDIT_LABEL                    32772
#define ID_EDIT_NEWLABEL                 32775
#define ID_EDIT_DELETE                   32776

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_3D_CONTROLS                  1
#define _APS_NEXT_RESOURCE_VALUE          135
#define _APS_NEXT_COMMAND_VALUE          32782
#define _APS_NEXT_CONTROL_VALUE           1002
#define _APS_NEXT_SYMED_VALUE             101
#endif
#endif
```

## RESPONSEEDIT.CPP;

```
// ResponseEdit.cpp : implementation file
//

#include "stdafx.h"
#include "TestReview.h"
#include "ResponseEdit.h"

#include "TestDoc.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CResponseEdit

CResponseEdit::CResponseEdit()
{
}

CResponseEdit::~CResponseEdit()
{
}

BEGIN_MESSAGE_MAP(CResponseEdit, CEdit)
    //{AFX_MSG_MAP(CResponseEdit)
    ON_CONTROL_REFLECT(EN_CHANGE, OnChange)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CResponseEdit message handlers

void CResponseEdit::OnChange()
// Notify the document that this edit control has changed
{
    (((CView*)GetParent())->GetDocument())->SetModifiedFlag();
}

```

## RESPONSEEDIT.H;

```
// ResponseEdit.h : header file

```

```
//
// CResponseEdit is an edit control that notifies it's view's underlying
// document when it is modified

#ifndef RESPONSEEDIT_H
#define RESPONSEEDIT_H

////////////////////////////////////
// CResponseEdit window

class CResponseEdit : public CEdit
{
// Construction
public:
    CResponseEdit();

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CResponseEdit)
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CResponseEdit();

    // Generated message map functions
protected:
    //{{AFX_MSG(CResponseEdit)
    afx_msg void OnChange();
    //}}AFX_MSG

    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

#endif // #ifndef RESPONSEEDIT_H
```

## STATICEDIT.CPP;

```
// StaticEdit.cpp : implementation file
//

#include "stdafx.h"
#include "TestReview.h"
#include "StaticEdit.h"
```



```

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CStaticEdit

CStaticEdit::CStaticEdit()
{
}

CStaticEdit::~CStaticEdit()
{
}

BEGIN_MESSAGE_MAP(CStaticEdit, CEdit)
    //{AFX_MSG_MAP(CStaticEdit)
    ON_WM_SETCURSOR()
    ON_WM_SETFOCUS()
    ON_WM_CHAR()
    ON_WM_RBUTTONDOWN()
    //}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CStaticEdit message handlers

BOOL CStaticEdit::OnSetCursor(CWnd* pWnd, UINT nHitTest, UINT message)
// Ignore the cursor
{
    // return CEdit::OnSetCursor(pWnd, nHitTest, message);
    return TRUE;
}

void CStaticEdit::OnSetFocus(CWnd* pOldWnd)
// Refuse to accept the input focus
{
    //CEdit::OnSetFocus(pOldWnd);

    pOldWnd->SetFocus();
}

void CStaticEdit::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
// Refuse to accept any characters from the keyboard
{
    //CEdit::OnChar(nChar, nRepCnt, nFlags);
}

void CStaticEdit::OnRButtonDown(UINT nFlags, CPoint point)
// Ignore this event
{
    //CEdit::OnRButtonDown(nFlags, point);
}

```

[illegible]

## STDAFX.CPP;

```
// stdafx.cpp : source file that includes just the standard includes
//     TestReview.pch will be the pre-compiled header
//     stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```

## STDAFX.H;

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
//     are changed infrequently
//

#define VC_EXTRALEAN          // Exclude rarely-used stuff from Windows
headers

#include <afxwin.h>           // MFC core and standard components
#include <afxext.h>           // MFC extensions
#include <afxole.h>           // MFC OLE classes
#include <afxodlgs.h>         // MFC OLE dialog classes
#include <afxcmn.h>           // MFC support for Windows 95 Common
Controls
#ifdef _AFX_NO_AFXCMN_SUPPORT
#endif // _AFX_NO_AFXCMN_SUPPORT

#include <afxdlgs.h>
```

## TESTDOC.CPP;

```
// TestDoc.cpp : implementation file
//

#include "stdafx.h"
#include "TestReview.h"
#include "TestDoc.h"
#include "DesignDoc.h"

#ifdef _DEBUG
```

```

#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CTestDoc

IMPLEMENT_DYNCREATE(CTestDoc, CDocument)

CTestDoc::CTestDoc()
{
}

BOOL CTestDoc::OnNewDocument()
// load a DesignDoc template and create a new TestDoc
{
    if (!CDocument::OnNewDocument())
        return FALSE;

    // Before creating a new TestDoc, load an existing design file into
    // a DesignDoc

    AfxMessageBox( "Please choose a question file (*.trq) to open",
                   MB_OK | MB_ICONINFORMATION );

    CFileDialog dlgOpen          // File Open Common Dialog
    ( TRUE,                      // file open
      "trq",                     // default extension
      "*.trq" );                // initial file name

    if ( dlgOpen.DoModal() != IDOK )
        // file name retrieval failed
        return FALSE;

    CDesignDoc design;
    design.OnOpenDocument( dlgOpen.GetPathName() );

    // Copy the DesignDoc info into the new TestDoc
    this->m_QRList.AddHead( &design.m_QRList );
    this->m_labelList.AddHead( &design.m_labelList );

    // Unassign the DesignDoc from the data
    design.m_QRList.RemoveAll();
    design.m_labelList.RemoveAll();

    return TRUE;
}

CTestDoc::~CTestDoc()
{
    // free the document data
    CQRData* pQRData;
    while ( !m_QRList.IsEmpty() ) {
        pQRData = (CQRData*) m_QRList.RemoveHead();
        delete pQRData;
    }
    m_QRList.RemoveAll();
}

```

```

        CEditData* pEditData;
        while ( !m_labelList.IsEmpty() ) {
            pEditData = (CEditData*) m_labelList.RemoveHead();
            delete pEditData;
        }
        m_labelList.RemoveAll();
    }

BEGIN_MESSAGE_MAP(CTestDoc, CDocument)
    //{AFX_MSG_MAP(CTestDoc)
        // NOTE - the ClassWizard will add and remove mapping macros here.
    //}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CTestDoc diagnostics

#ifdef _DEBUG
void CTestDoc::AssertValid() const
{
    CDocument::AssertValid();
}

void CTestDoc::Dump(CDumpContext& dc) const
{
    CDocument::Dump(dc);
}
#endif // _DEBUG

////////////////////////////////////
// CTestDoc serialization

void CTestDoc::Serialize(CArchive& ar)
{
    class CTestView;
    void UpdateTestDoc( CTestDoc* pDoc, CTestView* pView );

    if (ar.IsStoring())
        // Store data: first update the document data, then store it.
        {
            POSITION pos;
            UpdateTestDoc( this, (CTestView*) GetNextView(
pos=GetFirstViewPosition() ) );

            ar << m_QRList.GetCount();    // number of QRDatas in m_QRList

            // store each QRData from the list
            pos = m_QRList.GetHeadPosition();
            CQRData* pQRData;
            while ( pos ) {
                pQRData = (CQRData*) m_QRList.GetNext( pos );
                ar << pQRData->question.text << pQRData->question.rect;
                ar << pQRData->response.text << pQRData->response.rect;
            }

            ar << m_labelList.GetCount(); // number of EditDatas in
m_labelList

```



---

```

class CTestDoc : public CDocument
{
    friend class CTestView;
    friend void UpdateTestDoc( CTestDoc*, CTestView* );

protected:
    CPtrList    m_QRList;           // lists of data for question-response
pairs
    CPtrList    m_labellList;       // and labels respectively

    CTestDoc();                     // protected constructor used by dynamic creation
    DECLARE_DYNCREATE(CTestDoc)

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CTestDoc)
    public:
        virtual void Serialize(CArchive& ar);    // overridden for document i/o
    protected:
        virtual BOOL OnNewDocument();
    //{AFX_VIRTUAL

// Implementation
public:
    virtual ~CTestDoc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

    // Generated message map functions
protected:
    //{AFX_MSG(CTestDoc)
        // NOTE - the ClassWizard will add and remove member functions
here.
    //{AFX_MSG
    DECLARE_MESSAGE_MAP()
};

#endif          // ifndef TESTDOC_H

```

## TESTREVIEW.CPP;

```

// TestReview.cpp : Defines the class behaviors for the application.
//

```

```

#include "stdafx.h"
#include "TestReview.h"

#include "MainFrm.h"
#include "ChildFrm.h"
#include "DesignDoc.h"
#include "DesignView.h"
#include "TestDoc.h"
#include "TestView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CTestReviewApp

BEGIN_MESSAGE_MAP(CTestReviewApp, CWinApp)
//{{AFX_MSG_MAP(CTestReviewApp)
    ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        //      DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard file based document commands
    ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
    ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
// Standard print setup command
    ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

////////////////////////////////////
// CTestReviewApp construction

CTestReviewApp::CTestReviewApp()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

////////////////////////////////////
// The one and only CTestReviewApp object

CTestReviewApp theApp;

////////////////////////////////////
// CTestReviewApp initialization

BOOL CTestReviewApp::InitInstance()
{
    // Initialize OLE libraries
    if (!AfxOleInit())
    {
        AfxMessageBox(IDP_OLE_INIT_FAILED);
        return FALSE;
    }

    // Standard initialization

```



```

        // If you are not using these features and wish to reduce the size
        // of your final executable, you should remove from the following
        // the specific initialization routines you do not need.

#ifdef _AFXDLL
    Enable3dControls();           // Call this when using MFC in a
shared DLL
#else
    Enable3dControlsStatic();     // Call this when linking to MFC
statically
#endif

    LoadStdProfileSettings(); // Load standard INI file options (including
MRU)

    // Register the application's document templates. Document templates
    // serve as the connection between documents, frame windows and views.

    CMultiDocTemplate* pDocTemplate;

    // CDesignDoc template
    pDocTemplate = new CMultiDocTemplate(
        IDR_DESIGNTYPE,
        RUNTIME_CLASS(CDesignDoc),
        RUNTIME_CLASS(CChildFrame), // custom MDI child frame
        RUNTIME_CLASS(CDesignView));
    pDocTemplate->SetContainerInfo(IDR_DESIGNTYPE_CNTR_IP);
    AddDocTemplate(pDocTemplate);

    // CTestDoc template
    pDocTemplate = new CMultiDocTemplate(
        IDR_TESTTYPE,
        RUNTIME_CLASS(CTestDoc),
        RUNTIME_CLASS(CChildFrame), // custom MDI child frame
        RUNTIME_CLASS(CTestView));
    AddDocTemplate(pDocTemplate);

    // create main MDI Frame window
    CMainFrame* pMainFrame = new CMainFrame;
    if (!pMainFrame->LoadFrame(IDR_MAINFRAME))
        return FALSE;
    m_pMainWnd = pMainFrame;

    // Enable drag/drop open
    m_pMainWnd->DragAcceptFiles();

    // Enable DDE Execute open
    EnableShellOpen();
    RegisterShellFileTypes(TRUE);

    // The main window has been initialized, so show and update it.
    pMainFrame->ShowWindow(m_nCmdShow);
    pMainFrame->UpdateWindow();

    return TRUE;
}

```

```

////////////////////////////////////

```

[illegible]

## TESTREVIEW.H;

```
// TestReview.h : main header file for the TESTREVIEW application
//

#ifndef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"          // main symbols

////////////////////////////////////
// CTestReviewApp:
// See TestReview.cpp for the implementation of this class
//

class CTestReviewApp : public CWinApp
{
public:
    CTestReviewApp();

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CTestReviewApp)
public:
    virtual BOOL InitInstance();
    //}}AFX_VIRTUAL

// Implementation

    //{{AFX_MSG(CTestReviewApp)
    afx_msg void OnAppAbout();
    // NOTE - the ClassWizard will add and remove member functions
here.
        // DO NOT EDIT what you see in these blocks of generated code !
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////
```

## TESTREVIEW.RC;

```
//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
//
```

```

// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include \"afxres.h\"\\r\\n"
    "\\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "#define _AFX_NO_SPLITTER_RESOURCES\\r\\n"
    "#define _AFX_NO_PROPERTY_RESOURCES\\r\\n"
    "\\r\\n"
    "#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)\\r\\n"
    "#ifdef _WIN32\\r\\n"
    "LANGUAGE 9, 1\\r\\n"
    "#pragma code_page(1252)\\r\\n"
    "#endif\\r\\n"
    "#include \"res\\TestReview.rc2\" // non-Microsoft Visual C++ edited
resources\\r\\n"
    "#include \"afxres.rc\" // Standard components\\r\\n"
    "#include \"afxprint.rc\" // printing/print preview resources\\r\\n"
    "#include \"afxolecl.rc\" // OLE container resources\\r\\n"
    "#endif\\0"
END

#endif // APSTUDIO_INVOKED

////////////////////////////////////
//
// Icon
//

```

```

// Icon with lowest ID value placed first to ensure application icon
// remains consistent on all systems.
IDR_MAINFRAME          ICON      DISCARDABLE    "res\\TestReview.ico"
IDR_DESIGNTYPE         ICON      DISCARDABLE    "res\\DesignDoc.ico"
IDR_TESTTYPE           ICON      DISCARDABLE    "res\\TestDoc.ico"

////////////////////////////////////
//
// Bitmap
//

IDR_DESIGNTYPE         BITMAP  MOVEABLE PURE    "res\\mainfram.bmp"
IDR_TESTTYPE           BITMAP  MOVEABLE PURE    "res\\designnty.bmp"
IDR_MAINFRAME1         BITMAP  MOVEABLE PURE    "res\\bmp00001.bmp"
IDR_MAINFRAME          BITMAP  MOVEABLE PURE    "res\\testtype.bmp"

////////////////////////////////////
//
// Toolbar
//

IDR_DESIGNTYPE TOOLBAR DISCARDABLE  16, 15
BEGIN
    BUTTON        ID_FILE_NEW
    BUTTON        ID_FILE_OPEN
    BUTTON        ID_FILE_SAVE
    SEPARATOR
    BUTTON        ID_EDIT_CUT
    BUTTON        ID_EDIT_COPY
    BUTTON        ID_EDIT_PASTE
    SEPARATOR
    BUTTON        ID_FILE_PRINT
    BUTTON        ID_APP_ABOUT
    BUTTON        ID_CONTEXT_HELP
END

IDR_TESTTYPE TOOLBAR DISCARDABLE  16, 15
BEGIN
    BUTTON        ID_FILE_NEW
    BUTTON        ID_FILE_OPEN
    BUTTON        ID_FILE_SAVE
    SEPARATOR
    BUTTON        ID_EDIT_CUT
    BUTTON        ID_EDIT_COPY
    BUTTON        ID_EDIT_PASTE
    SEPARATOR
    BUTTON        ID_FILE_PRINT
    BUTTON        ID_APP_ABOUT
    BUTTON        ID_CONTEXT_HELP
    BUTTON        ID_EDIT_NEWQR
    BUTTON        ID_EDIT_NEWLABEL
END

IDR_MAINFRAME1 TOOLBAR DISCARDABLE  16, 15
BEGIN
    BUTTON        ID_FILE_NEW
    BUTTON        ID_FILE_OPEN
    BUTTON        ID_FILE_SAVE
    SEPARATOR

```

```

        BUTTON      ID_FILE_PRINT
        BUTTON      ID_APP_ABOUT
        BUTTON      ID_CONTEXT_HELP
END

```

```
IDR_MAINFRAME TOOLBAR DISCARDABLE 16, 15
```

```

BEGIN
    BUTTON      ID_FILE_NEW
    BUTTON      ID_FILE_OPEN
    BUTTON      ID_FILE_SAVE
    SEPARATOR
    BUTTON      ID_EDIT_NEWQR
    BUTTON      ID_EDIT_NEWLABEL
    SEPARATOR
    BUTTON      ID_EDIT_DELETE
    SEPARATOR
    BUTTON      ID_APP_ABOUT
END

```

```

////////////////////////////////////
//
// Menu
//

```

```
IDR_MAINFRAME MENU PRELOAD DISCARDABLE
```

```

BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New...\tCtrl+N",          ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",          ID_FILE_OPEN
        MENUITEM SEPARATOR
        MENUITEM "Recent File",                ID_FILE_MRU_FILE1, GRAYED
        MENUITEM SEPARATOR
        MENUITEM "E&xit",                      ID_APP_EXIT
    END
    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",                  ID_VIEW_TOOLBAR
        MENUITEM "&Status Bar",               ID_VIEW_STATUS_BAR
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&About TestReview...",      ID_APP_ABOUT
    END
END

```

```
IDR_DESIGNTYPE MENU PRELOAD DISCARDABLE
```

```

BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New...\tCtrl+N",          ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",          ID_FILE_OPEN
        MENUITEM "&Close",                    ID_FILE_CLOSE
        MENUITEM "&Save\tCtrl+S",              ID_FILE_SAVE
        MENUITEM "Save &As...",                ID_FILE_SAVE_AS
        MENUITEM SEPARATOR
        MENUITEM "Recent File",                ID_FILE_MRU_FILE1, GRAYED
        MENUITEM SEPARATOR
    END

```

```

        MENUITEM "E&xit",                ID_APP_EXIT
    END
    POPUP "&Edit"
    BEGIN
        MENUITEM "New &Question-Response", ID_EDIT_NEWQR
        MENUITEM "New &Label",             ID_EDIT_NEWLABEL
        MENUITEM SEPARATOR
        MENUITEM "&Delete",                ID_EDIT_DELETE
    END
    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",              ID_VIEW_TOOLBAR
        MENUITEM "&Status Bar",           ID_VIEW_STATUS_BAR
    END
    POPUP "&Window"
    BEGIN
        MENUITEM "&New Window",           ID_WINDOW_NEW
        MENUITEM "&Cascade",              ID_WINDOW_CASCADE
        MENUITEM "&Tile",                  ID_WINDOW_TILE_HORZ
        MENUITEM "&Arrange Icons",        ID_WINDOW_ARRANGE
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&About TestReview...", ID_APP_ABOUT
    END
END

IDR_DESIGNTYPE_CNTR_IP MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",          ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",      ID_FILE_OPEN
        MENUITEM "&Close",                 ID_FILE_CLOSE
        MENUITEM "&Save\tCtrl+S",          ID_FILE_SAVE
        MENUITEM "Save &As...",            ID_FILE_SAVE_AS
        MENUITEM SEPARATOR
        MENUITEM "&Print...\tCtrl+P",      ID_FILE_PRINT
        MENUITEM "Print Pre&view",         ID_FILE_PRINT_PREVIEW
        MENUITEM "P&rint Setup...",        ID_FILE_PRINT_SETUP
        MENUITEM SEPARATOR
        MENUITEM "Recent File",            ID_FILE_MRU_FILE1, GRAYED
        MENUITEM SEPARATOR
        MENUITEM "E&xit",                  ID_APP_EXIT
    END
    MENUITEM SEPARATOR
    MENUITEM SEPARATOR
    POPUP "&Window"
    BEGIN
        MENUITEM "&New Window",           ID_WINDOW_NEW
        MENUITEM "&Cascade",              ID_WINDOW_CASCADE
        MENUITEM "&Tile",                  ID_WINDOW_TILE_HORZ
        MENUITEM "&Arrange Icons",        ID_WINDOW_ARRANGE
    END
END

IDR_TESTTYPE MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"

```

```

BEGIN
    MENUITEM "&New...\tCtrl+N",          ID_FILE_NEW
    MENUITEM "&Open...\tCtrl+O",        ID_FILE_OPEN
    MENUITEM "&Close",                  ID_FILE_CLOSE
    MENUITEM "&Save\tCtrl+S",            ID_FILE_SAVE
    MENUITEM "Save &As...",              ID_FILE_SAVE_AS
    MENUITEM SEPARATOR
    MENUITEM "Recent File",              ID_FILE_MRU_FILE1, GRAYED
    MENUITEM SEPARATOR
    MENUITEM "E&xit",                    ID_APP_EXIT
END
POPUP "&View"
BEGIN
    MENUITEM "&Toolbar",                ID_VIEW_TOOLBAR
    MENUITEM "&Status Bar",            ID_VIEW_STATUS_BAR
END
POPUP "&Window"
BEGIN
    MENUITEM "&New Window",             ID_WINDOW_NEW
    MENUITEM "&Cascade",                ID_WINDOW_CASCADE
    MENUITEM "&Tile",                  ID_WINDOW_TILE_HORZ
    MENUITEM "&Arrange Icons",          ID_WINDOW_ARRANGE
END
POPUP "&Help"
BEGIN
    MENUITEM "&About TestReview...",    ID_APP_ABOUT
END
END

IDR_TESTTYPE1 MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",        ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",    ID_FILE_OPEN
        MENUITEM "&Close",              ID_FILE_CLOSE
        MENUITEM "&Save\tCtrl+S",        ID_FILE_SAVE
        MENUITEM "Save &As...",          ID_FILE_SAVE_AS
        MENUITEM SEPARATOR
        MENUITEM "&Print...\tCtrl+P",    ID_FILE_PRINT
        MENUITEM "Print Pre&view",        ID_FILE_PRINT_PREVIEW
        MENUITEM "P&rint Setup...",      ID_FILE_PRINT_SETUP
        MENUITEM SEPARATOR
        MENUITEM "Recent File",          ID_FILE_MRU_FILE1, GRAYED
        MENUITEM SEPARATOR
        MENUITEM "E&xit",                ID_APP_EXIT
    END
    POPUP "&Edit"
    BEGIN
        MENUITEM "&Undo\tCtrl+Z",        ID_EDIT_UNDO
        MENUITEM SEPARATOR
        MENUITEM "Cu&t\tCtrl+X",          ID_EDIT_CUT
        MENUITEM "&Copy\tCtrl+C",        ID_EDIT_COPY
        MENUITEM "&Paste\tCtrl+V",        ID_EDIT_PASTE
    END
    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",            ID_VIEW_TOOLBAR
        MENUITEM "&Status Bar",        ID_VIEW_STATUS_BAR
    END

```



```

END
POPUP "&Window"
BEGIN
    MENUITEM "&New Window",           ID_WINDOW_NEW
    MENUITEM "&Cascade",             ID_WINDOW_CASCADE
    MENUITEM "&Tile",                 ID_WINDOW_TILE_HORZ
    MENUITEM "&Arrange Icons",        ID_WINDOW_ARRANGE
END
POPUP "&Help"
BEGIN
    MENUITEM "&Help Topics",          ID_HELP_FINDER
    MENUITEM SEPARATOR
    MENUITEM "&About TestReview...", ID_APP_ABOUT
END
END

IDR_MAINFRAME2 MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",      ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",  ID_FILE_OPEN
        MENUITEM SEPARATOR
        MENUITEM "Recent File",        ID_FILE_MRU_FILE1, GRAYED
        MENUITEM SEPARATOR
        MENUITEM "E&xit",              ID_APP_EXIT
    END
    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",           ID_VIEW_TOOLBAR
        MENUITEM "&Status Bar",        ID_VIEW_STATUS_BAR
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&Help Topics",          ID_HELP_FINDER
        MENUITEM SEPARATOR
        MENUITEM "&About TestReview...", ID_APP_ABOUT
    END
END

IDR_DESIGNTYPE_CNTR_IP1 MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",      ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",  ID_FILE_OPEN
        MENUITEM "&Close",             ID_FILE_CLOSE
        MENUITEM "&Save\tCtrl+S",      ID_FILE_SAVE
        MENUITEM "Save &As...",        ID_FILE_SAVE_AS
        MENUITEM SEPARATOR
        MENUITEM "&Print...\tCtrl+P",  ID_FILE_PRINT
        MENUITEM "Print Pre&view",      ID_FILE_PRINT_PREVIEW
        MENUITEM "P&rint Setup...",     ID_FILE_PRINT_SETUP
        MENUITEM SEPARATOR
        MENUITEM "Recent File",        ID_FILE_MRU_FILE1, GRAYED
        MENUITEM SEPARATOR
        MENUITEM "E&xit",              ID_APP_EXIT
    END
    MENUITEM SEPARATOR

```

```

MENUITEM SEPARATOR
POPUP "&Window"
BEGIN
    MENUITEM "&New Window",
    MENUITEM "&Cascade",
    MENUITEM "&Tile",
    MENUITEM "&Arrange Icons",
END
END

ID_WINDOW_NEW
ID_WINDOW_CASCADE
ID_WINDOW_TILE_HORZ
ID_WINDOW_ARRANGE

IDR_DESIGNTYPE1 MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",
        MENUITEM "&Open...\tCtrl+O",
        MENUITEM "&Close",
        MENUITEM "&Save\tCtrl+S",
        MENUITEM "Save &As...",
        MENUITEM SEPARATOR
        MENUITEM "&Print...\tCtrl+P",
        MENUITEM "Print Pre&view",
        MENUITEM "P&rint Setup...",
        MENUITEM SEPARATOR
        MENUITEM "Recent File",
        MENUITEM SEPARATOR
        MENUITEM "E&xit",
    END
    ID_FILE_NEW
    ID_FILE_OPEN
    ID_FILE_CLOSE
    ID_FILE_SAVE
    ID_FILE_SAVE_AS
    ID_FILE_PRINT
    ID_FILE_PRINT_PREVIEW
    ID_FILE_PRINT_SETUP
    ID_FILE_MRU_FILE1, GRAYED
    ID_APP_EXIT

    POPUP "&Edit"
    BEGIN
        MENUITEM "&Undo\tCtrl+Z",
        MENUITEM SEPARATOR
        MENUITEM "Cu&t\tCtrl+X",
        MENUITEM "&Copy\tCtrl+C",
        MENUITEM "&Paste\tCtrl+V",
        MENUITEM SEPARATOR
        MENUITEM "New &Question-Response",
        MENUITEM "New &Label",
    END
    ID_EDIT_UNDO
    ID_EDIT_CUT
    ID_EDIT_COPY
    ID_EDIT_PASTE
    ID_EDIT_NEWQR
    ID_EDIT_NEWLABEL

    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",
        MENUITEM "&Status Bar",
    END
    ID_VIEW_TOOLBAR
    ID_VIEW_STATUS_BAR

    POPUP "&Window"
    BEGIN
        MENUITEM "&New Window",
        MENUITEM "&Cascade",
        MENUITEM "&Tile",
        MENUITEM "&Arrange Icons",
    END
    ID_WINDOW_NEW
    ID_WINDOW_CASCADE
    ID_WINDOW_TILE_HORZ
    ID_WINDOW_ARRANGE

    POPUP "&Help"
    BEGIN
        MENUITEM "&Help Topics",
        MENUITEM SEPARATOR
        MENUITEM "&About TestReview...",
    END
    ID_HELP_FINDER
    ID_APP_ABOUT
END

```

```

////////////////////////////////////
//
// Accelerator
//

```

```

IDR_MAINFRAME ACCELERATORS PRELOAD MOVEABLE PURE
BEGIN

```

"C",	ID_EDIT_COPY,	VIRTKEY, CONTROL, NOINVERT
"N",	ID_FILE_NEW,	VIRTKEY, CONTROL, NOINVERT
"O",	ID_FILE_OPEN,	VIRTKEY, CONTROL, NOINVERT
"P",	ID_FILE_PRINT,	VIRTKEY, CONTROL, NOINVERT
"S",	ID_FILE_SAVE,	VIRTKEY, CONTROL, NOINVERT
"V",	ID_EDIT_PASTE,	VIRTKEY, CONTROL, NOINVERT
VK_BACK,	ID_EDIT_UNDO,	VIRTKEY, ALT, NOINVERT
VK_DELETE,	ID_EDIT_CUT,	VIRTKEY, SHIFT, NOINVERT
VK_ESCAPE,	ID_CANCEL_EDIT_CNTR,	VIRTKEY, NOINVERT
VK_F6,	ID_NEXT_PANE,	VIRTKEY, NOINVERT
VK_F6,	ID_PREV_PANE,	VIRTKEY, SHIFT, NOINVERT
VK_INSERT,	ID_EDIT_COPY,	VIRTKEY, CONTROL, NOINVERT
VK_INSERT,	ID_EDIT_PASTE,	VIRTKEY, SHIFT, NOINVERT
"X",	ID_EDIT_CUT,	VIRTKEY, CONTROL, NOINVERT
"Z",	ID_EDIT_UNDO,	VIRTKEY, CONTROL, NOINVERT

```

END

```

```

IDR_DESIGNTYPE_CNTR_IP ACCELERATORS PRELOAD MOVEABLE PURE
BEGIN

```

"N",	ID_FILE_NEW,	VIRTKEY, CONTROL, NOINVERT
"O",	ID_FILE_OPEN,	VIRTKEY, CONTROL, NOINVERT
"P",	ID_FILE_PRINT,	VIRTKEY, CONTROL, NOINVERT
"S",	ID_FILE_SAVE,	VIRTKEY, CONTROL, NOINVERT
VK_ESCAPE,	ID_CANCEL_EDIT_CNTR,	VIRTKEY, NOINVERT
VK_F6,	ID_NEXT_PANE,	VIRTKEY, NOINVERT
VK_F6,	ID_PREV_PANE,	VIRTKEY, SHIFT, NOINVERT

```

END

```

```

IDR_MAINFRAME3 ACCELERATORS PRELOAD MOVEABLE PURE
BEGIN

```

"N",	ID_FILE_NEW,	VIRTKEY, CONTROL
"O",	ID_FILE_OPEN,	VIRTKEY, CONTROL
"S",	ID_FILE_SAVE,	VIRTKEY, CONTROL
"P",	ID_FILE_PRINT,	VIRTKEY, CONTROL
"Z",	ID_EDIT_UNDO,	VIRTKEY, CONTROL
"X",	ID_EDIT_CUT,	VIRTKEY, CONTROL
"C",	ID_EDIT_COPY,	VIRTKEY, CONTROL
"V",	ID_EDIT_PASTE,	VIRTKEY, CONTROL
VK_BACK,	ID_EDIT_UNDO,	VIRTKEY, ALT
VK_DELETE,	ID_EDIT_CUT,	VIRTKEY, SHIFT
VK_INSERT,	ID_EDIT_COPY,	VIRTKEY, CONTROL
VK_INSERT,	ID_EDIT_PASTE,	VIRTKEY, SHIFT
VK_F6,	ID_NEXT_PANE,	VIRTKEY
VK_F6,	ID_PREV_PANE,	VIRTKEY, SHIFT
VK_F1,	ID_CONTEXT_HELP,	VIRTKEY, SHIFT
VK_F1,	ID_HELP,	VIRTKEY
VK_ESCAPE,	ID_CANCEL_EDIT_CNTR,	VIRTKEY, NOINVERT

```

END

```

```

IDR_DESIGNTYPE_CNTR_IP2 ACCELERATORS PRELOAD MOVEABLE PURE
BEGIN

```

```

        "N",          ID_FILE_NEW,          VIRTKEY, CONTROL
        "O",          ID_FILE_OPEN,        VIRTKEY, CONTROL
        "S",          ID_FILE_SAVE,        VIRTKEY, CONTROL
        "P",          ID_FILE_PRINT,       VIRTKEY, CONTROL
        VK_F6,        ID_NEXT_PANE,        VIRTKEY
        VK_F6,        ID_PREV_PANE,        VIRTKEY, SHIFT
        VK_F1,        ID_CONTEXT_HELP,     VIRTKEY, SHIFT
        VK_F1,        ID_HELP,            VIRTKEY
        VK_ESCAPE,    ID_CANCEL_EDIT_CNTR, VIRTKEY, NOINVERT
END

```

```

////////////////////////////////////
//
// Dialog
//

```

```

IDD_ABOUTBOX DIALOG DISCARDABLE 0, 0, 203, 108
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "About TestReview"
FONT 8, "MS Sans Serif"
BEGIN
    ICON            IDR_MAINFRAME, IDC_STATIC, 26, 43, 16, 16
    LTEXT           "TestReview Version 1.0", IDC_STATIC, 58, 29, 80, 8,
                    SS_NOPREFIX
    LTEXT           "Copyright © 1996", IDC_STATIC, 65, 65, 59, 8
    DEFPUSHBUTTON   "OK", IDOK, 146, 46, 32, 14, WS_GROUP
END

```

```

#ifndef _MAC
////////////////////////////////////
//
// Version
//

```

```

VS_VERSION_INFO VERSIONINFO
    FILEVERSION 1,0,0,1
    PRODUCTVERSION 1,0,0,1
    FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
    FILEFLAGS 0x1L
#else
    FILEFLAGS 0x0L
#endif
    FILEOS 0x4L
    FILETYPE 0x1L
    FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "000004b0"
        BEGIN
            VALUE "Comments", "This is Chitu's very first Visual C++
program\0"
            VALUE "CompanyName", "ISDS Lab Student\0"
            VALUE "FileDescription", "TestReview\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TestReview\0"

```

```
        VALUE "LegalCopyright", "Copyright © 1996\0"
        VALUE "OriginalFilename", "TestReview.exe\0"
        VALUE "ProductName", "TestReview\0"
        VALUE "ProductVersion", "1, 0, 0, 1\0"
    END
END
BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x0, 1200
END
END

#endif    // !_MAC

////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_ABOUTBOX, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 196
        TOPMARGIN, 7
        BOTTOMMARGIN, 101
    END
END
#endif    // APSTUDIO_INVOKED

////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDP_OLE_INIT_FAILED    "OLE initialization failed.  Make sure that the
OLE libraries are the correct version."
    IDP_FAILED_TO_CREATE    "Failed to create object.  Make sure the object is
entered in the system registry."
END

STRINGTABLE PRELOAD DISCARDABLE
BEGIN
    IDR_MAINFRAME            "TestReview"
    IDR_DESIGNTYPE            "\nQuestionSet\nQuestion\nTestReview Question
Files (*.trq)\n.trq\nTestReviewQuestion.Document\nTestReview Question
Document"
    IDR_TESTTYPE              "\nTest\nTest\nTestReview Test Files
(*.trt)\n.trt\nTestReview.Test.Document\nTestReview Test Document"
END

STRINGTABLE PRELOAD DISCARDABLE
BEGIN
```

```

        AFX_IDS_APP_TITLE           "TestReview"
        AFX_IDS_IDLEMESSAGE         " "
        AFX_IDS_HELPMODEMESSAGE    "Select an object on which to get Help"
    END

    STRINGTABLE DISCARDABLE
    BEGIN
        ID_INDICATOR_EXT            "EXT"
        ID_INDICATOR_CAPS           "CAP"
        ID_INDICATOR_NUM            "NUM"
        ID_INDICATOR_SCRL           "SCRL"
        ID_INDICATOR_OVR            "OVR"
        ID_INDICATOR_REC            "REC"
    END

    STRINGTABLE DISCARDABLE
    BEGIN
        ID_FILE_NEW                 "Create a new document\nNew"
        ID_FILE_OPEN                 "Open an existing document\nOpen"
        ID_FILE_CLOSE                "Close the active document\nClose"
        ID_FILE_SAVE                 "Save the active document\nSave"
        ID_FILE_SAVE_AS              "Save the active document with a new name\nSave
    As"
        ID_FILE_PAGE_SETUP           "Change the printing options\nPage Setup"
        ID_FILE_PRINT_SETUP          "Change the printer and printing options\nPrint
    Setup"
        ID_FILE_PRINT                "Print the active document\nPrint"
        ID_FILE_PRINT_PREVIEW        "Display full pages\nPrint Preview"
    END

    STRINGTABLE DISCARDABLE
    BEGIN
        ID_APP_ABOUT                 "Display program information, version number and
    copyright\nAbout"
        ID_APP_EXIT                  "Quit the application; prompts to save
    documents\nExit"
        ID_HELP_INDEX                "Opens Help\nHelp Topics"
        ID_HELP_FINDER               "List Help topics\nHelp Topics"
        ID_HELP_USING                "Display instructions about how to use help\nHelp"
        ID_CONTEXT_HELP              "Display help for clicked on buttons, menus and
    windows\nHelp"
        ID_HELP                      "Display help for current task or command\nHelp"
    END

    STRINGTABLE DISCARDABLE
    BEGIN
        ID_FILE_MRU_FILE1            "Open this document"
        ID_FILE_MRU_FILE2            "Open this document"
        ID_FILE_MRU_FILE3            "Open this document"
        ID_FILE_MRU_FILE4            "Open this document"
        ID_FILE_MRU_FILE5            "Open this document"
        ID_FILE_MRU_FILE6            "Open this document"
        ID_FILE_MRU_FILE7            "Open this document"
        ID_FILE_MRU_FILE8            "Open this document"
        ID_FILE_MRU_FILE9            "Open this document"
        ID_FILE_MRU_FILE10           "Open this document"
        ID_FILE_MRU_FILE11           "Open this document"
        ID_FILE_MRU_FILE12           "Open this document"
        ID_FILE_MRU_FILE13           "Open this document"
    END

```

```

        ID_FILE_MRU_FILE14      "Open this document"
        ID_FILE_MRU_FILE15      "Open this document"
        ID_FILE_MRU_FILE16      "Open this document"
END

STRINGTABLE DISCARDABLE
BEGIN
        ID_NEXT_PANE            "Switch to the next window pane\nNext Pane"
        ID_PREV_PANE            "Switch back to the previous window pane\nPrevious
Pane"
END

STRINGTABLE DISCARDABLE
BEGIN
        ID_WINDOW_NEW            "Open another window for the active document\nNew
Window"
        ID_WINDOW_ARRANGE        "Arrange icons at the bottom of the
window\nArrange Icons"
        ID_WINDOW_CASCADE        "Arrange windows so they overlap\nCascade Windows"
        ID_WINDOW_TILE_HORZ      "Arrange windows as non-overlapping tiles\nTile
Windows"
        ID_WINDOW_TILE_VERT      "Arrange windows as non-overlapping tiles\nTile
Windows"
        ID_WINDOW_SPLIT          "Split the active window into panes\nSplit"
END

STRINGTABLE DISCARDABLE
BEGIN
        ID_EDIT_CLEAR            "Erase the selection\nErase"
        ID_EDIT_CLEAR_ALL        "Erase everything\nErase All"
        ID_EDIT_COPY             "Copy the selection and put it on the
Clipboard\nCopy"
        ID_EDIT_CUT              "Cut the selection and put it on the
Clipboard\nCut"
        ID_EDIT_FIND             "Find the specified text\nFind"
        ID_EDIT_PASTE            "Insert Clipboard contents\nPaste"
        ID_EDIT_PASTE_LINK        "Insert Clipboard contents and a link to its
source\nPaste Link"
        ID_EDIT_PASTE_SPECIAL    "Insert Clipboard contents with options\nPaste
Special"
        ID_EDIT_REPEAT           "Repeat the last action\nRepeat"
        ID_EDIT_REPLACE          "Replace specific text with different
text\nReplace"
        ID_EDIT_SELECT_ALL       "Select the entire document\nSelect All"
        ID_EDIT_UNDO             "Undo the last action\nUndo"
        ID_EDIT_REDO             "Redo the previously undone action\nRedo"
END

STRINGTABLE DISCARDABLE
BEGIN
        ID_VIEW_TOOLBAR          "Show or hide the toolbar\nToggle ToolBar"
        ID_VIEW_STATUS_BAR       "Show or hide the status bar\nToggle StatusBar"
END

STRINGTABLE DISCARDABLE
BEGIN
        ID_OLE_INSERT_NEW        "Insert new embedded object\nNew Object"
        ID_OLE_EDIT_LINKS        "Edit linked objects\nEdit Links"
        ID_OLE_EDIT_CONVERT      "Convert object to different type\nConvert Object"

```

END

STRINGTABLE DISCARDABLE  
BEGIN

ID_OLE_VERB_FIRST	"Activate embedded or linked object"
57873	"Activate embedded or linked object"
57874	"Activate embedded or linked object"
57875	"Activate embedded or linked object"

END

STRINGTABLE DISCARDABLE  
BEGIN

AFX_IDS_SCSIZE	"Change the window size"
AFX_IDS_SCMOVE	"Change the window position"
AFX_IDS_SCMINIMIZE	"Reduce the window to an icon"
AFX_IDS_SCMAXIMIZE	"Enlarge the window to full size"
AFX_IDS_SCNEXTWINDOW	"Switch to the next document window"
AFX_IDS_SCPREVWINDOW	"Switch to the previous document window"
AFX_IDS_SCCLOSE	"Close the active window and prompts to save the documents"

END

STRINGTABLE DISCARDABLE  
BEGIN

AFX_IDS_SCRESTORE	"Restore the window to normal size"
AFX_IDS_SCTASKLIST	"Activate Task List"
AFX_IDS_MDICHILD	"Activate this window"

END

STRINGTABLE DISCARDABLE  
BEGIN

AFX_IDS_PREVIEW_CLOSE	"Close print preview mode\nCancel Preview"
-----------------------	--

END

STRINGTABLE DISCARDABLE  
BEGIN

ID_EDIT_NEWQR	"Create a new Question-Response pair\nNew Question-Response"
ID_EDIT_LABEL	"Create a new label\nNew Label"
ID_EDIT_NEWLABEL	"Create a new label\nNew Label"
ID_EDIT_DELETE	"Delete a label or question-response pair\nDelete"

END

#endif // English (U.S.) resources

////////////////////////////////////

#ifndef APSTUDIO\_INVOKED

////////////////////////////////////  
//

// Generated from the TEXTINCLUDE 3 resource.

//

#define \_AFX\_NO\_SPLITTER\_RESOURCES

#define \_AFX\_NO\_PROPERTY\_RESOURCES

#if !defined(AFX\_RESOURCE\_DLL) || defined(AFX\_TARG\_ENU)

#ifdef WIN32

LANGUAGE 9, 1



```
#pragma code_page(1252)
#endif
#include "res\TestReview.rc2" // non-Microsoft Visual C++ edited resources
#include "afxres.rc"         // Standard components
#include "afxprint.rc"       // printing/print preview resources
#include "afxolecl.rc"       // OLE container resources
#endif
////////////////////////////////////
#endif // not APSTUDIO_INVOKED
```

## TESTREVIEW.RC2;

```
//
// TESTREVIEW.RC2 - resources Microsoft Visual C++ does not edit directly
//

#ifdef APSTUDIO_INVOKED
    #error this file is not editable by Microsoft Visual C++
#endif //APSTUDIO_INVOKED

////////////////////////////////////
// Add manually edited resources here...
////////////////////////////////////
```

## TESTVIEW.CPP;

```
// TestView.cpp : implementation file
//

#include "stdafx.h"
#include "TestReview.h"
#include "TestView.h"
#include "ResponseEdit.h"
#include "StaticEdit.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CTestView
```

---

```

IMPLEMENT_DYNCREATE(CTestView, CScrollView)

CTestView::CTestView()
{
}

CTestView::~~CTestView()
{
    // free the view data
    CStatic* pStatic;
    while ( !m_questionList.IsEmpty() ) {
        pStatic = (CStatic*) m_questionList.RemoveHead();
        delete pStatic;
    }
    m_questionList.RemoveAll();

    CEdit* pEdit;
    while ( !m_responseList.IsEmpty() ) {
        pEdit = (CEdit*) m_responseList.RemoveHead();
        delete pEdit;
    }
    m_responseList.RemoveAll();

    while ( !m_labelList.IsEmpty() ) {
        pStatic = (CStatic*) m_labelList.RemoveHead();
        delete pStatic;
    }
    m_labelList.RemoveAll();
}

BEGIN_MESSAGE_MAP(CTestView, CScrollView)
    //{{AFX_MSG_MAP(CTestView)
        // NOTE - the ClassWizard will add and remove mapping macros here.
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CTestView drawing

void CTestView::OnInitialUpdate()
{
    CScrollView::OnInitialUpdate();

    CSize sizeTotal;
    // TODO: calculate the total size of this view
    sizeTotal.cx = 850;
    sizeTotal.cy = 10000;
    SetScrollSizes(MM_TEXT, sizeTotal);

    CTestDoc* pDoc = (CTestDoc*) GetDocument();

    // Construct any existing questions and responses from the document
data.
    CStaticEdit* pQuestion;
    CResponseEdit* pResponse;
    CQRData* pQRData;
    for ( POSITION pos = pDoc->m_QRList.GetHeadPosition(); pos != NULL; ) {

```

```

        pQRData = (CQRData*) pDoc->m_QRList.GetNext( pos );

        m_questionList.AddTail( pQuestion = new CStaticEdit );
        m_responseList.AddTail( pResponse = new CResponseEdit );

        pQuestion->Create(          WS_CHILD | WS_VISIBLE | ES_MULTILINE,
                                   pQRData->question.rect, this, AssignID()
);
        pQuestion->SetWindowText( pQRData->question.text );
        pResponse->Create(          WS_CHILD | WS_VISIBLE | WS_BORDER |
WS_TABSTOP |
                                   ES_MULTILINE | ES_WANTRETURN |
                                   WS_VSCROLL,
                                   pQRData->response.rect, this,
AssignID() );
        pResponse->SetWindowText( pQRData->response.text );
    }

    // Construct any existing labels from the document data.
    CStaticEdit* pLabel;
    CEditData* pEditData;
    for ( pos = pDoc->m_labelList.GetHeadPosition(); pos != NULL; ) {
        pEditData = (CEditData*) pDoc->m_labelList.GetNext( pos );

        this->m_labelList.AddTail( pLabel = new CStaticEdit );

        pLabel->Create(    WS_CHILD | WS_VISIBLE | ES_MULTILINE,
                           pEditData->rect, this, AssignID() );
        pLabel->SetWindowText( pEditData->text );
    }

    // don't set off a false alarm
    pDoc->SetModifiedFlag( FALSE );
}

void CTestView::OnDraw(CDC* pDC)
{
    CDocument* pDoc = GetDocument();
    // TODO: add draw code here
}

////////////////////////////////////
// CTestView diagnostics

#ifdef _DEBUG
void CTestView::AssertValid() const
{
    CScrollView::AssertValid();
}

void CTestView::Dump(CDumpContext& dc) const
{
    CScrollView::Dump(dc);
}
#endif // _DEBUG

////////////////////////////////////

```

```
// CTestView message handlers
```

## TESTVIEW.H.

```
// TestView.h : header file
//

////////////////////////////////////
// CTestView view

#ifndef TESTVIEW_H
#define TESTVIEW_H

#include "TestDoc.h"
#include "EditStruct.h"
#include "ControlID.h"

class CTestView : public CScrollView, protected CControlID
{
    friend void UpdateTestDoc( CTestDoc*, CTestView* );

protected:
    CTestView();          // protected constructor used by dynamic creation
    DECLARE_DYNCREATE(CTestView)

// Attributes
public:

protected:
    CObList      m_labelList;
    CObList      m_questionList;
    CObList      m_responseList;

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CTestView)
protected:
    virtual void OnDraw(CDC* pDC);          // overridden to draw this view
    virtual void OnInitialUpdate();        // first time after construct
    //}}AFX_VIRTUAL

// Implementation
protected:
    virtual ~CTestView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

    // Generated message map functions
```

[illegible]