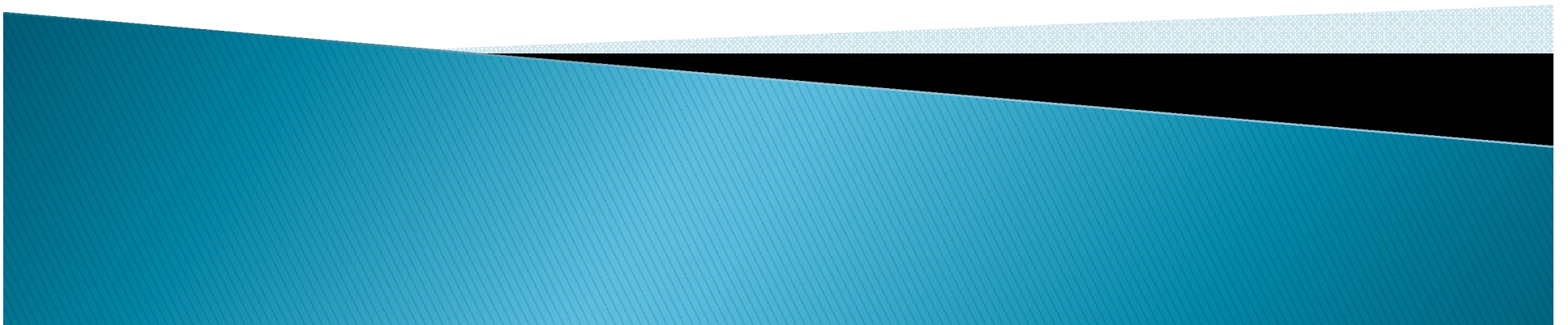
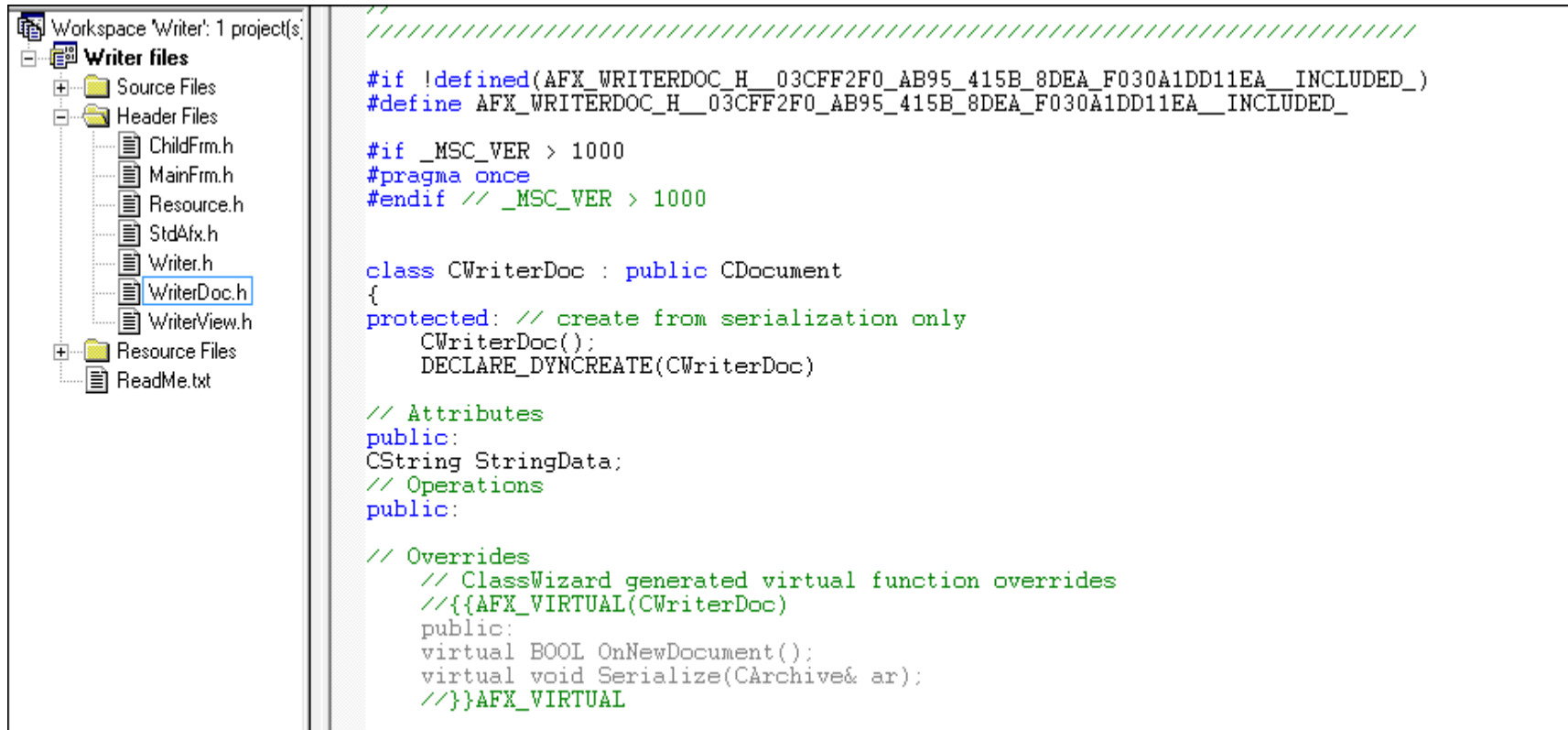


Serializer

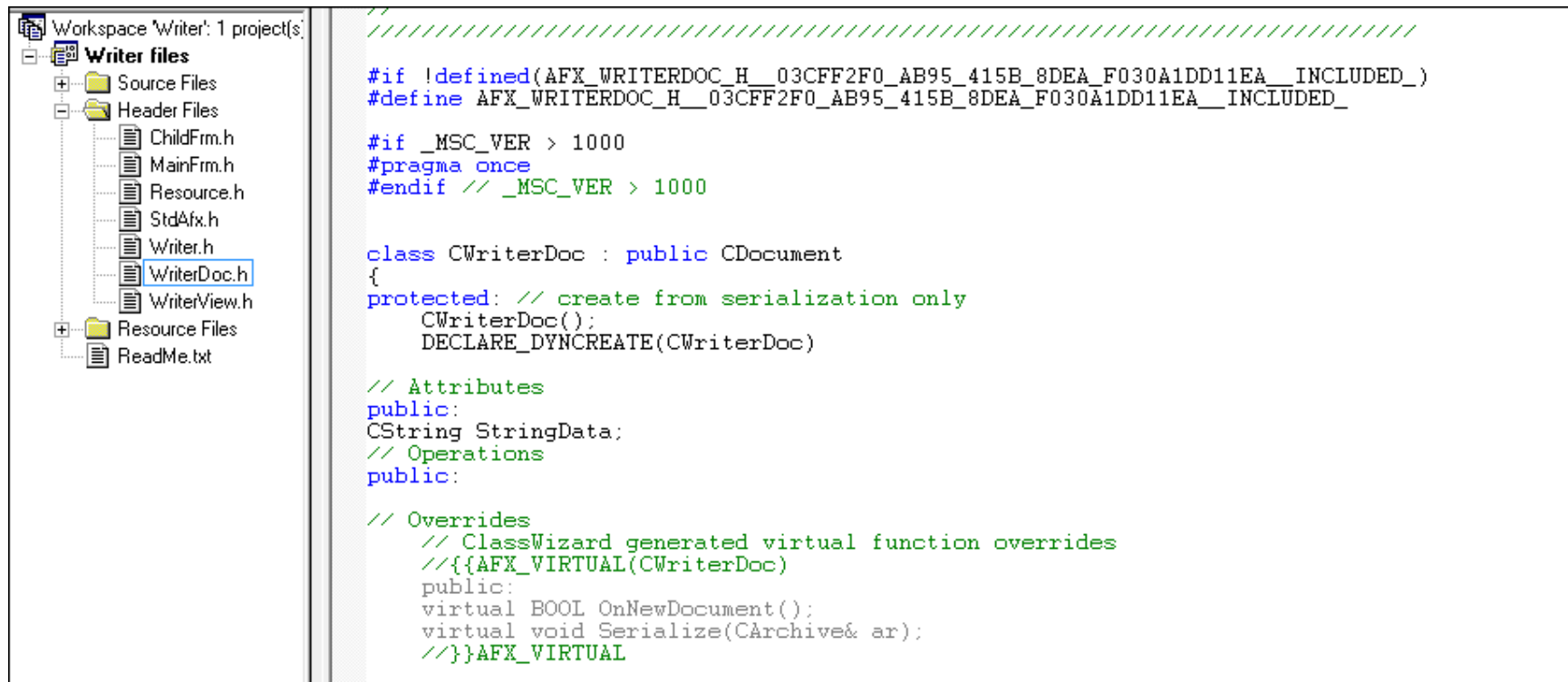


WriterDoc.h



```
//  
////////////////////////////////////  
#if !defined(AFX_WRITERDOC_H_03CFF2F0_AB95_415B_8DEA_F030A1DD11EA__INCLUDED_)  
#define AFX_WRITERDOC_H_03CFF2F0_AB95_415B_8DEA_F030A1DD11EA__INCLUDED_  
  
#if _MSC_VER > 1000  
#pragma once  
#endif // _MSC_VER > 1000  
  
class CWriterDoc : public CDocument  
{  
protected: // create from serialization only  
    CWriterDoc();  
    DECLARE_DYNCREATE(CWriterDoc)  
  
// Attributes  
public:  
    CString StringData;  
// Operations  
public:  
  
// Overrides  
    // ClassWizard generated virtual function overrides  
    //{{AFX_VIRTUAL(CWriterDoc)  
    public:  
    virtual BOOL OnNewDocument();  
    virtual void Serialize(CArchive& ar);  
    //}}AFX_VIRTUAL
```

WriterDoc.h

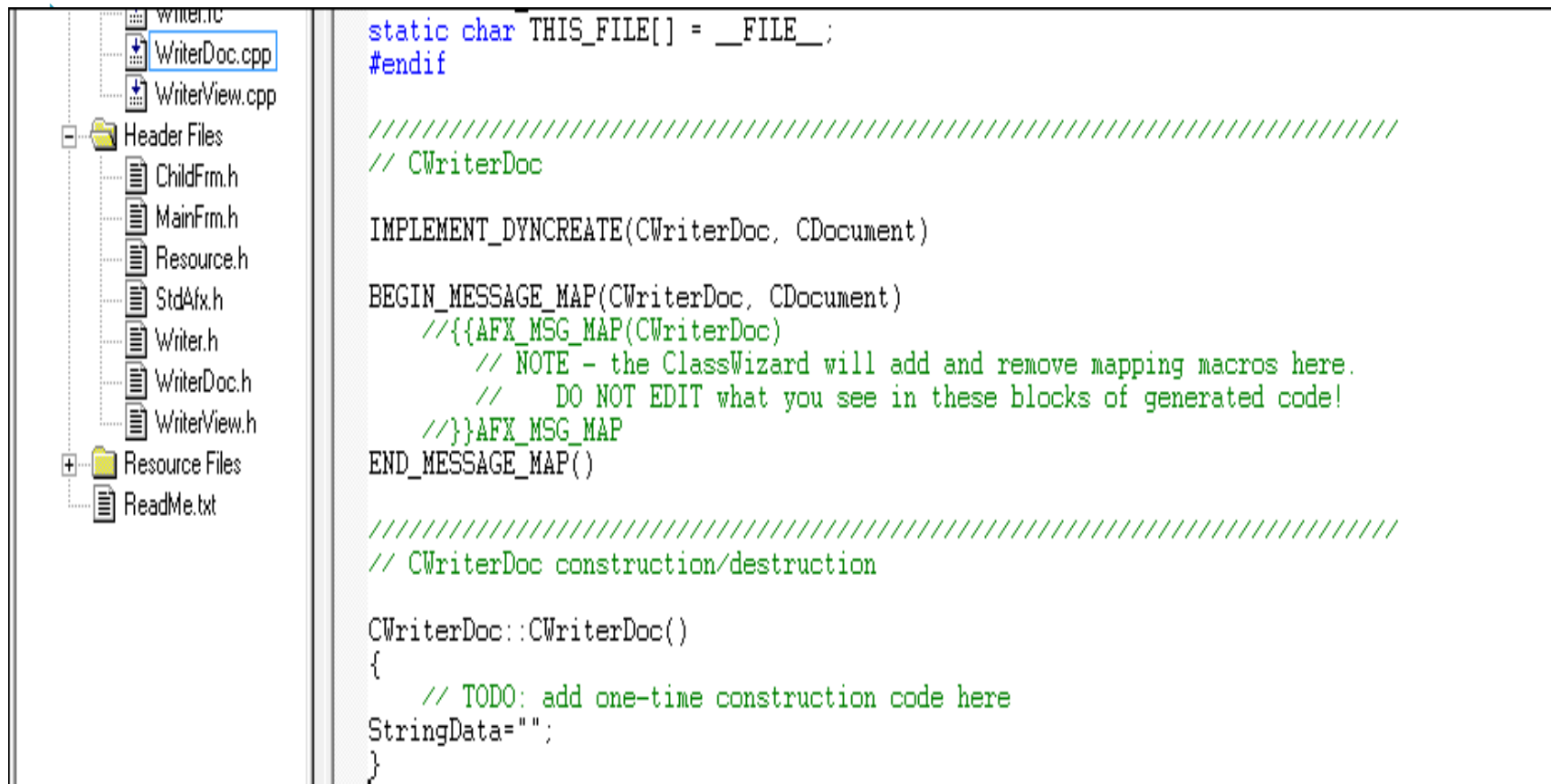


The image shows a screenshot of an IDE. On the left is a file explorer for a workspace named 'Writer'. It shows a project structure with 'Source Files' and 'Header Files' folders. Under 'Header Files', there are several files: ChildFrm.h, MainFrm.h, Resource.h, StdAfx.h, Writer.h, WriterDoc.h (which is selected), and WriterView.h. Below these are 'Resource Files' and a 'ReadMe.txt' file.

The main editor area displays the code for WriterDoc.h. The code is as follows:

```
////////////////////////////////////  
#if !defined(AFX_WRITERDOC_H_03CFF2F0_AB95_415B_8DEA_F030A1DD11EA__INCLUDED_)  
#define AFX_WRITERDOC_H_03CFF2F0_AB95_415B_8DEA_F030A1DD11EA__INCLUDED_  
  
#if _MSC_VER > 1000  
#pragma once  
#endif // _MSC_VER > 1000  
  
class CWriterDoc : public CDocument  
{  
protected: // create from serialization only  
    CWriterDoc();  
    DECLARE_DYNCREATE(CWriterDoc)  
  
// Attributes  
public:  
    CString StringData;  
// Operations  
public:  
  
// Overrides  
// ClassWizard generated virtual function overrides  
//{{AFX_VIRTUAL(CWriterDoc)  
public:  
    virtual BOOL OnNewDocument();  
    virtual void Serialize(CArchive& ar);  
//}}AFX_VIRTUAL
```

WriterDoc.cpp



The image shows a screenshot of a code editor. On the left is a file explorer showing a project structure with folders for 'Header Files' and 'Resource Files'. The 'Header Files' folder contains files like ChildFrm.h, MainFrm.h, Resource.h, StdAfx.h, Writer.h, WriterDoc.h, and WriterView.h. The 'Resource Files' folder contains ReadMe.txt. The main editor area displays the source code for WriterDoc.cpp, which includes file identification, class implementation macros, message map, and the constructor.

```
static char THIS_FILE[] = __FILE__;
#endif

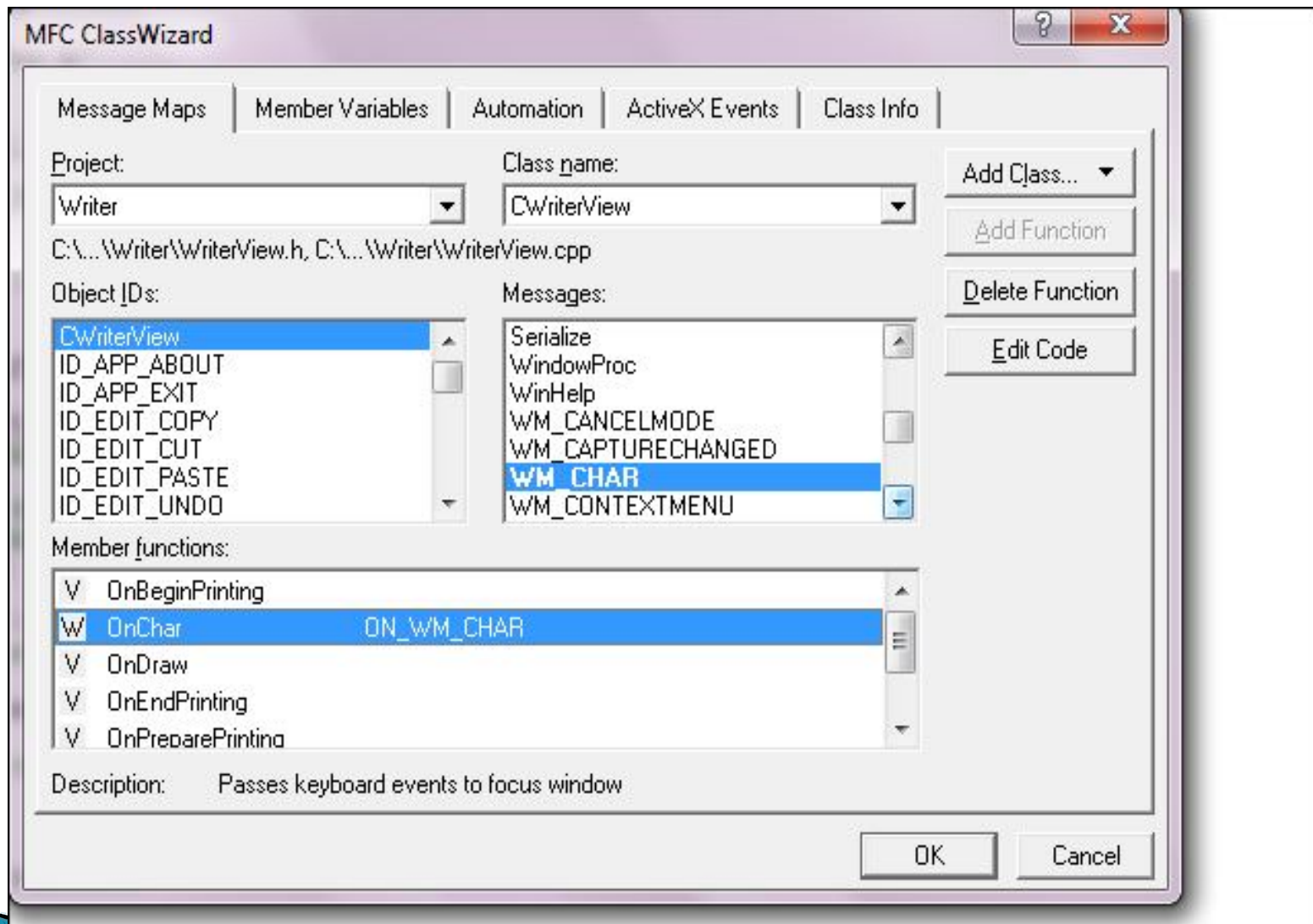
////////////////////////////////////
// CWriterDoc

IMPLEMENT_DYNCREATE(CWriterDoc, CDocument)

BEGIN_MESSAGE_MAP(CWriterDoc, CDocument)
    //{AFX_MSG_MAP(CWriterDoc)
    // 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()

////////////////////////////////////
// CWriterDoc construction/destruction

CWriterDoc::CWriterDoc()
{
    // TODO: add one-time construction code here
    StringData="";
}
```



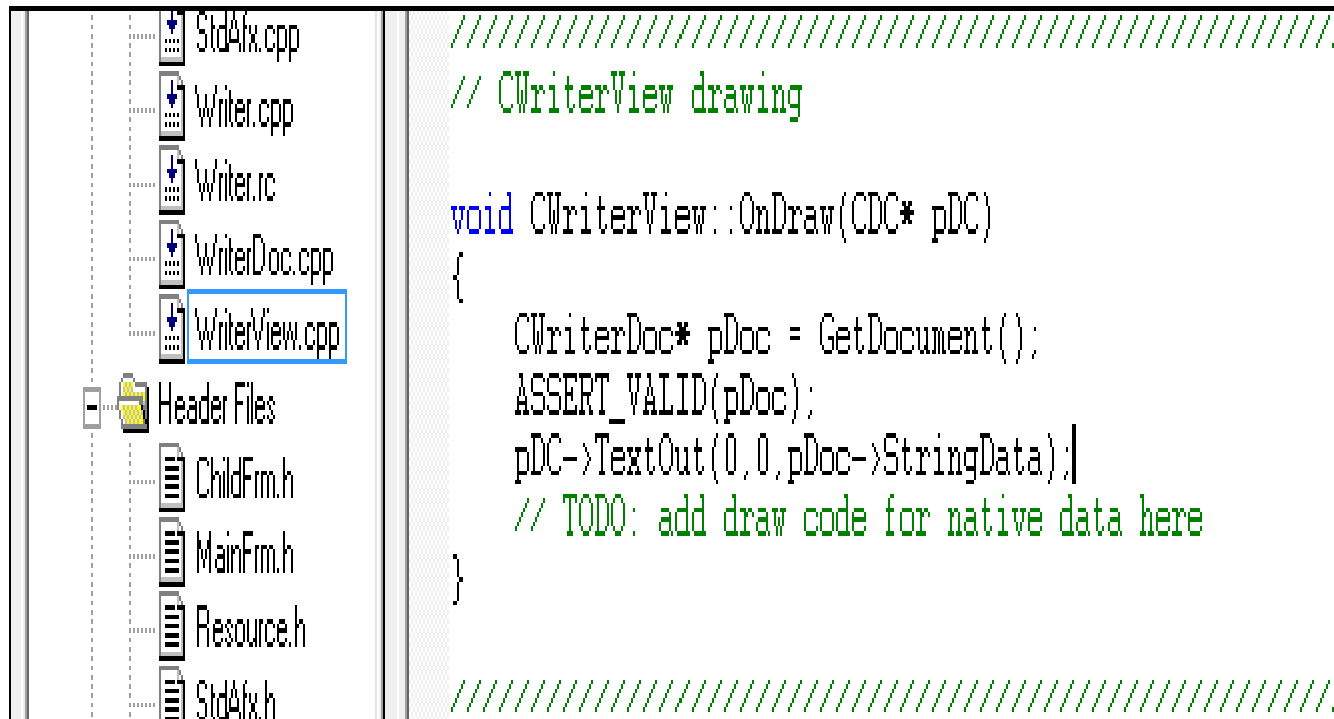
```
Writer.cpp
Writer.rc
WriterDoc.cpp
WriterView.cpp
Header Files
  ChildFrm.h
  MainFrm.h
  Resource.h
  StdAfx.h
  Writer.h
  WriterDoc.h
  WriterView.h
Resource Files
  ReadMe.txt

////////////////////////////////////
// CWriterView message handlers

void CWriterView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: Add your message handler code here and/or call default

    CWriterDoc* pDoc=GetDocument();
    ASSERT_VALID(pDoc);
    pDoc->StringData +=nChar;
    Invalidate();
    CView::OnChar(nChar, nRepCnt, nFlags);
}
```

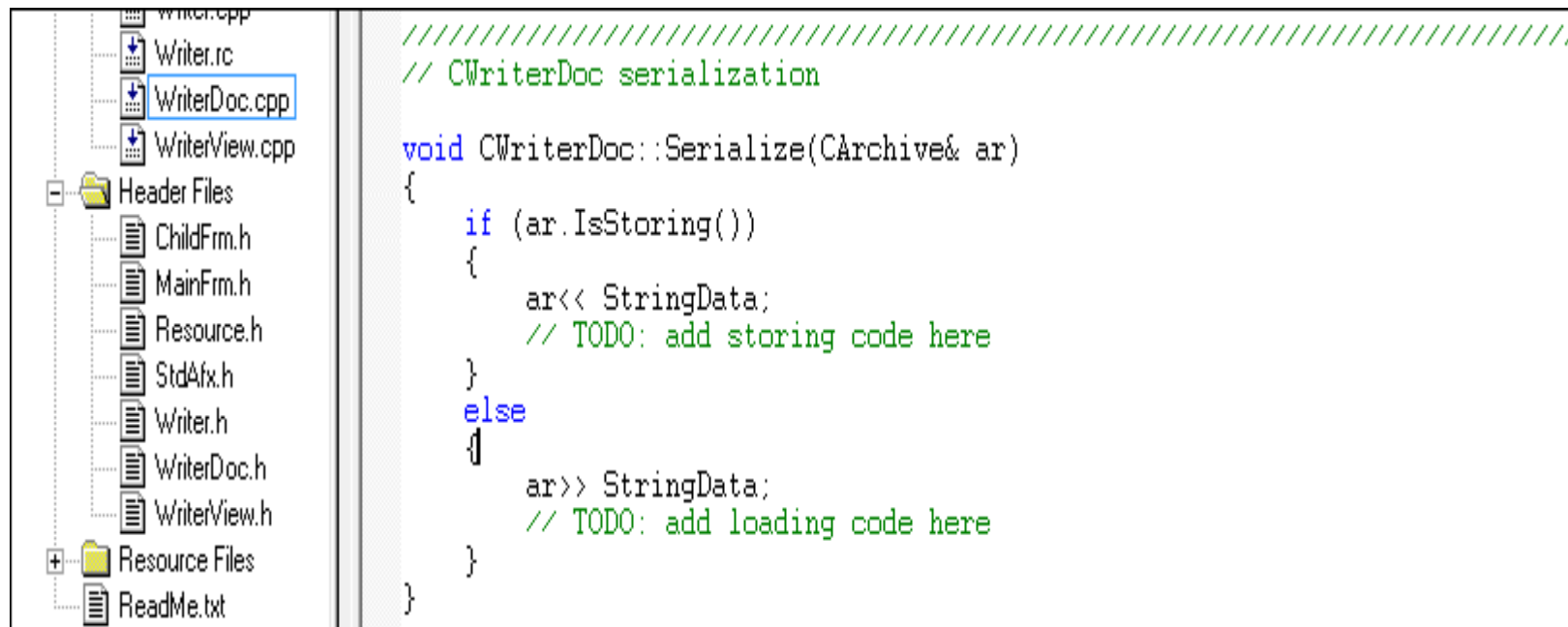
WriterView.cpp



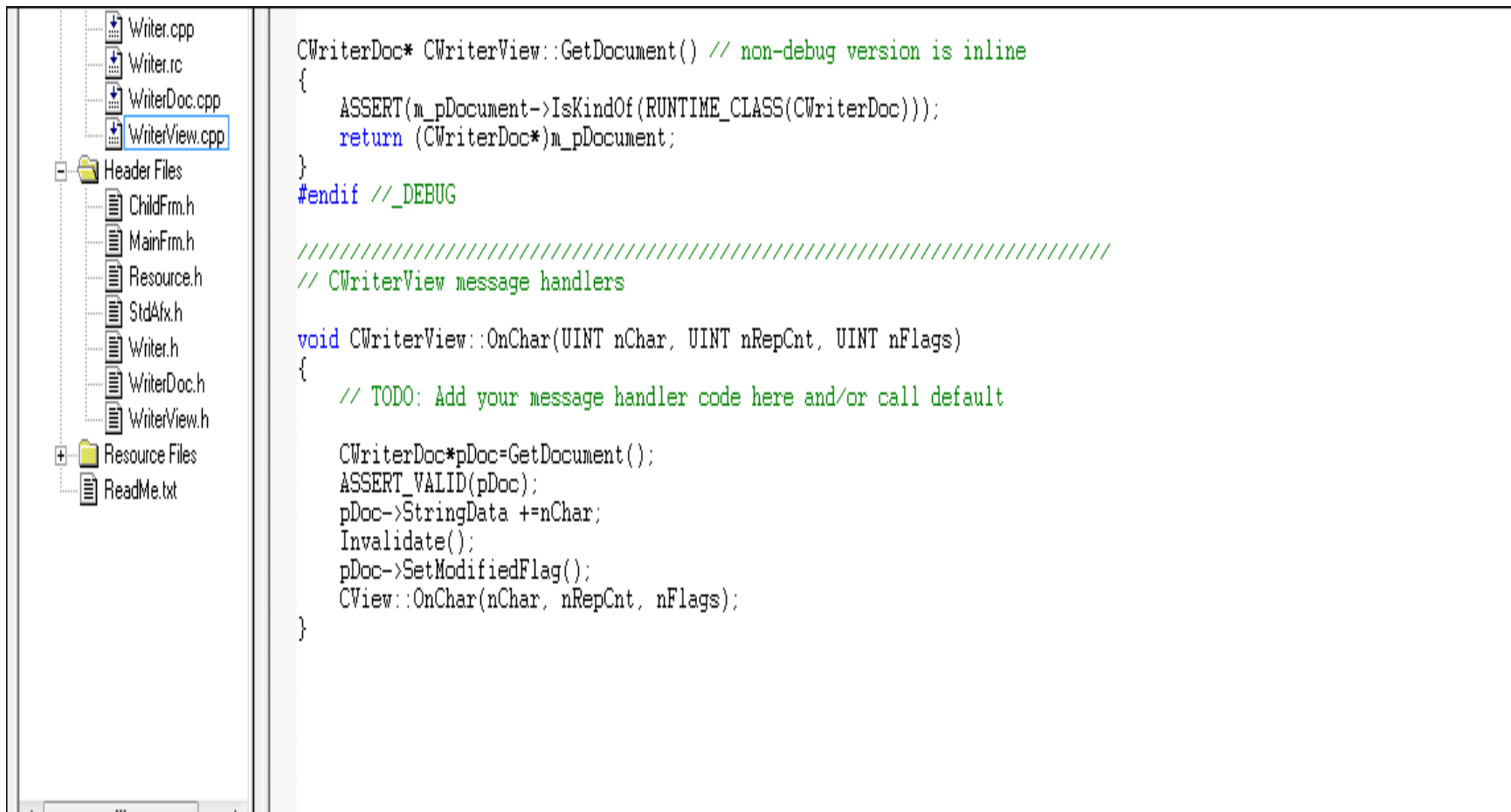
```
//////  
// CWriterView drawing  
  
void CWriterView::OnDraw(CDC* pDC)  
{  
    CWriterDoc* pDoc = GetDocument();  
    ASSERT_VALID(pDoc);  
    pDC->TextOut(0, 0, pDoc->StringData);  
    // TODO: add draw code for native data here  
}  
  
//////
```

The screenshot shows a code editor with a project file tree on the left. The tree includes files like StdAfx.cpp, Writer.cpp, Writer.rc, WriterDoc.cpp, and WriterView.cpp (which is selected). Under 'Header Files', there are ChildFrm.h, MainFrm.h, Resource.h, and StdAfx.h. The main editor area displays the implementation of the OnDraw method for CWriterView, which currently only outputs the document's string data to the device context.

writerDoc.cpp



```
////////////////////////////////////  
// CWriterDoc serialization  
  
void CWriterDoc::Serialize(CArchive& ar)  
{  
    if (ar.IsStoring())  
    {  
        ar<< StringData;  
        // TODO: add storing code here  
    }  
    else  
    {  
        ar>> StringData;  
        // TODO: add loading code here  
    }  
}
```

```
Writer.cpp
Writer.rc
WriterDoc.cpp
WriterView.cpp
Header Files
ChildFrm.h
MainFrm.h
Resource.h
StdAfx.h
Writer.h
WriterDoc.h
WriterView.h
Resource Files
ReadMe.txt

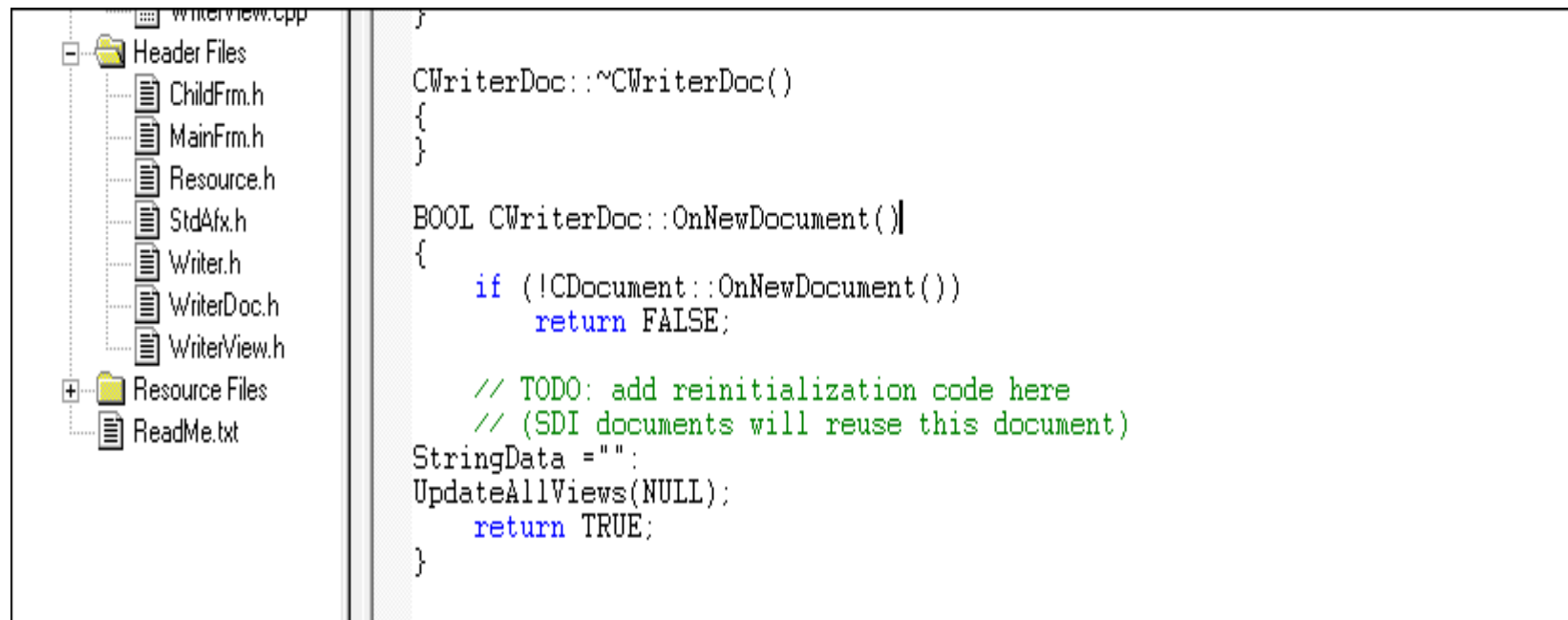
CWriterDoc* CWriterView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CWriterDoc)));
    return (CWriterDoc*)m_pDocument;
}
#endif // _DEBUG

////////////////////////////////////
// CWriterView message handlers

void CWriterView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: Add your message handler code here and/or call default

    CWriterDoc*pDoc=GetDocument();
    ASSERT_VALID(pDoc);
    pDoc->StringData +=nChar;
    Invalidate();
    pDoc->SetModifiedFlag();
    CView::OnChar(nChar, nRepCnt, nFlags);
}
```

WriterDoc.cpp



```

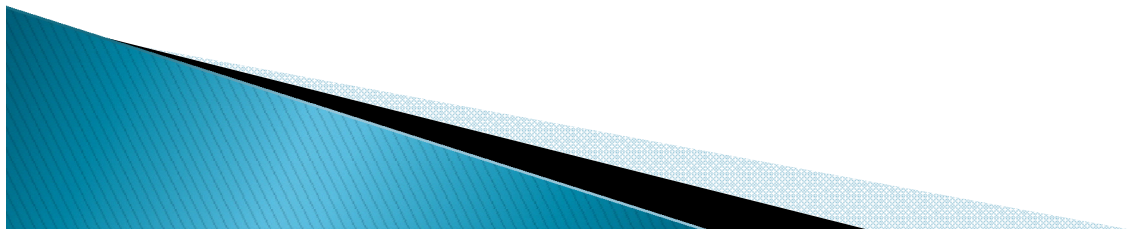
}

CWriterDoc::~CWriterDoc()
{
}

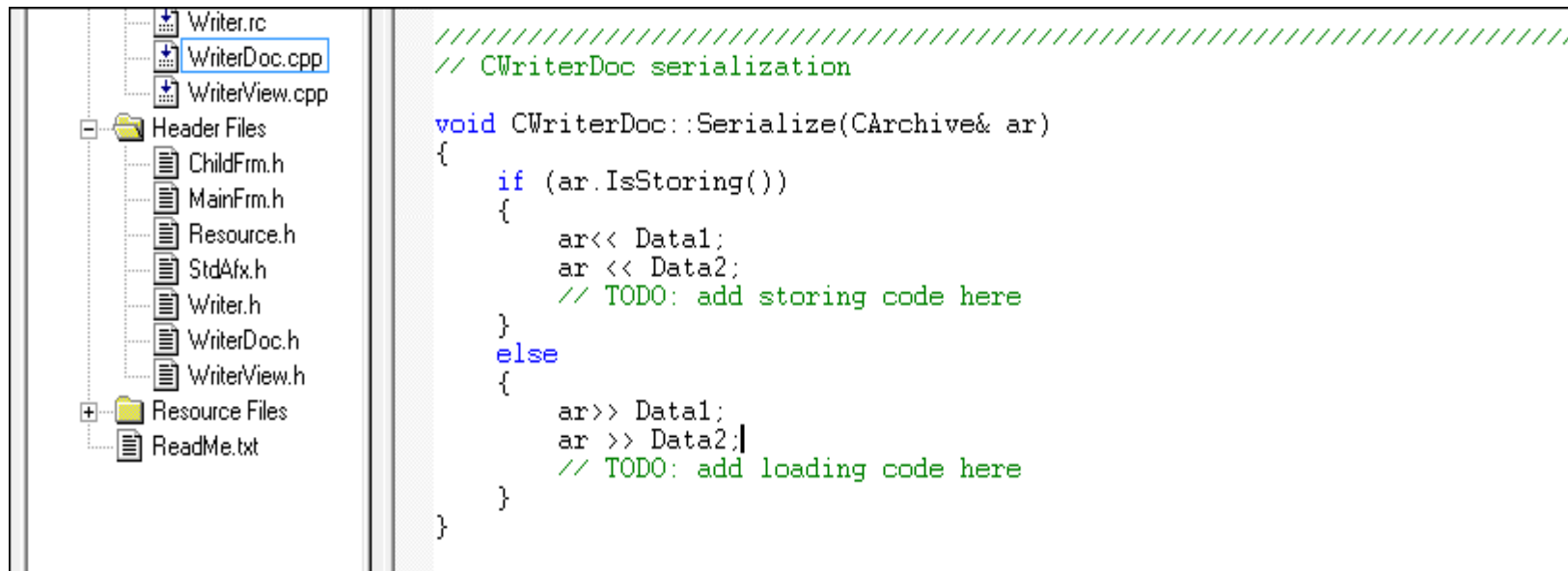
BOOL CWriterDoc::OnNewDocument()
{
    if (!CDocument::OnNewDocument())
        return FALSE;

    // TODO: add reinitialization code here
    // (SDI documents will reuse this document)
    StringData = "";
    UpdateAllViews(NULL);
    return TRUE;
}

```



WriterDoc.cpp



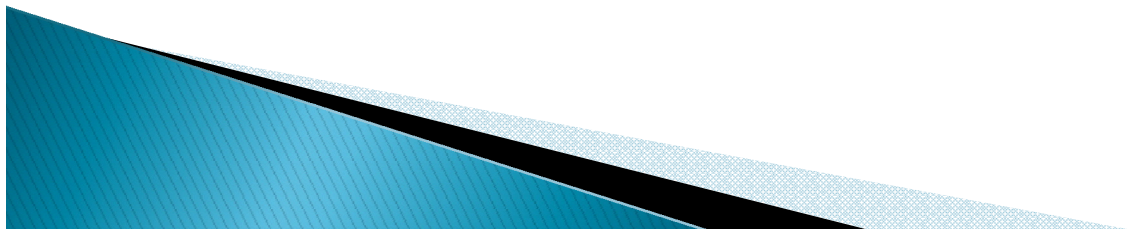
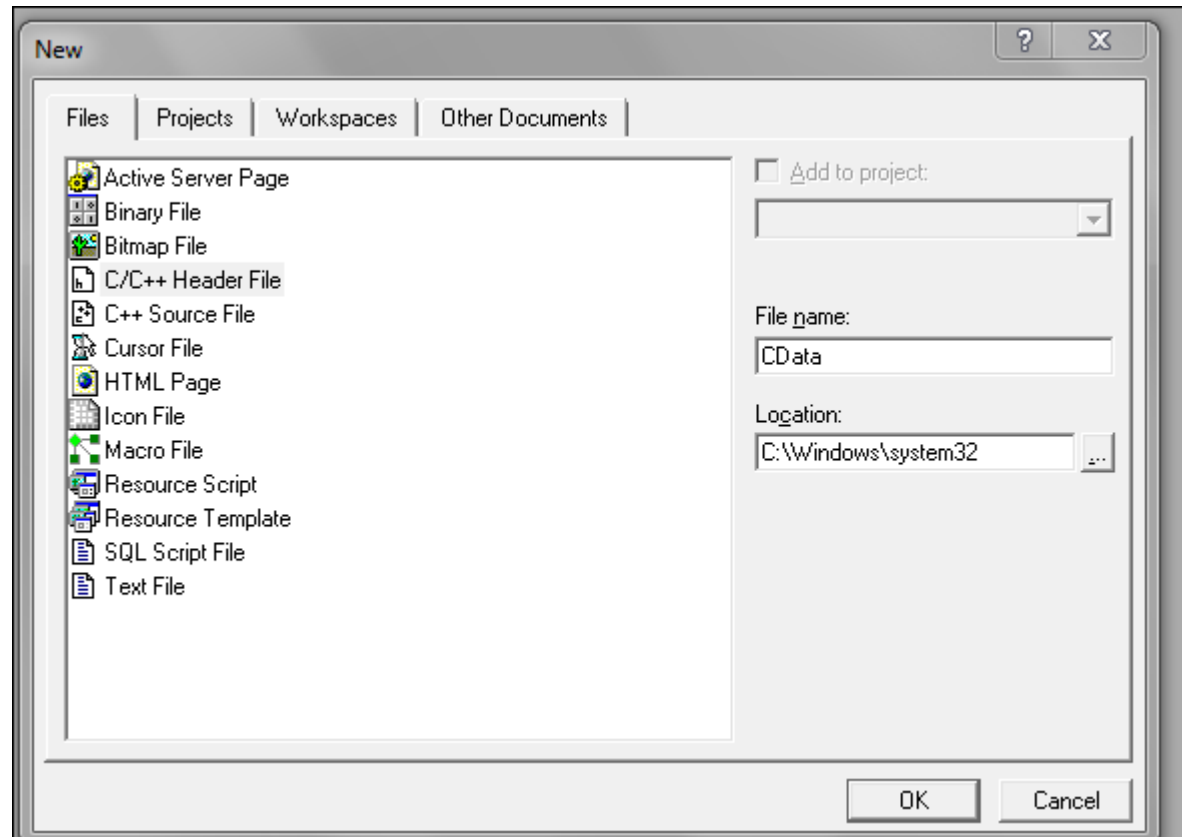
The image shows a screenshot of a code editor. On the left is a project file tree with the following structure:

- Writer.rc
- WriterDoc.cpp (highlighted)
- WriterView.cpp
- Header Files
 - ChildFrm.h
 - MainFrm.h
 - Resource.h
 - StdAfx.h
 - Writer.h
 - WriterDoc.h
 - WriterView.h
- Resource Files
- ReadMe.txt

On the right, the code for `CWriterDoc::Serialize` is shown:

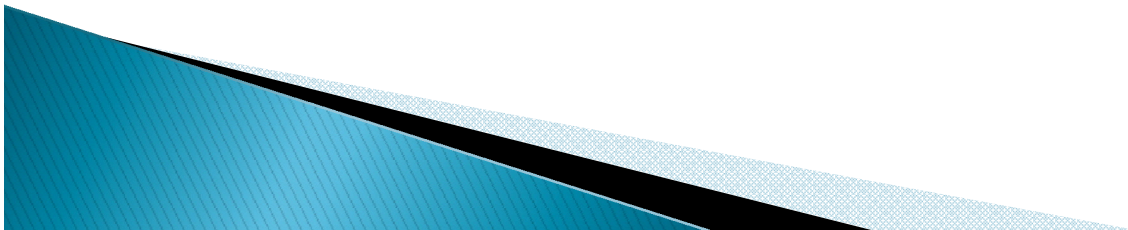
```
////////////////////////////////////  
// CWriterDoc serialization  
  
void CWriterDoc::Serialize(CArchive& ar)  
{  
    if (ar.IsStoring())  
    {  
        ar<< Data1;  
        ar << Data2;  
        // TODO: add storing code here  
    }  
    else  
    {  
        ar>> Data1;  
        ar >> Data2;  
        // TODO: add loading code here  
    }  
}
```

Serializing our own class

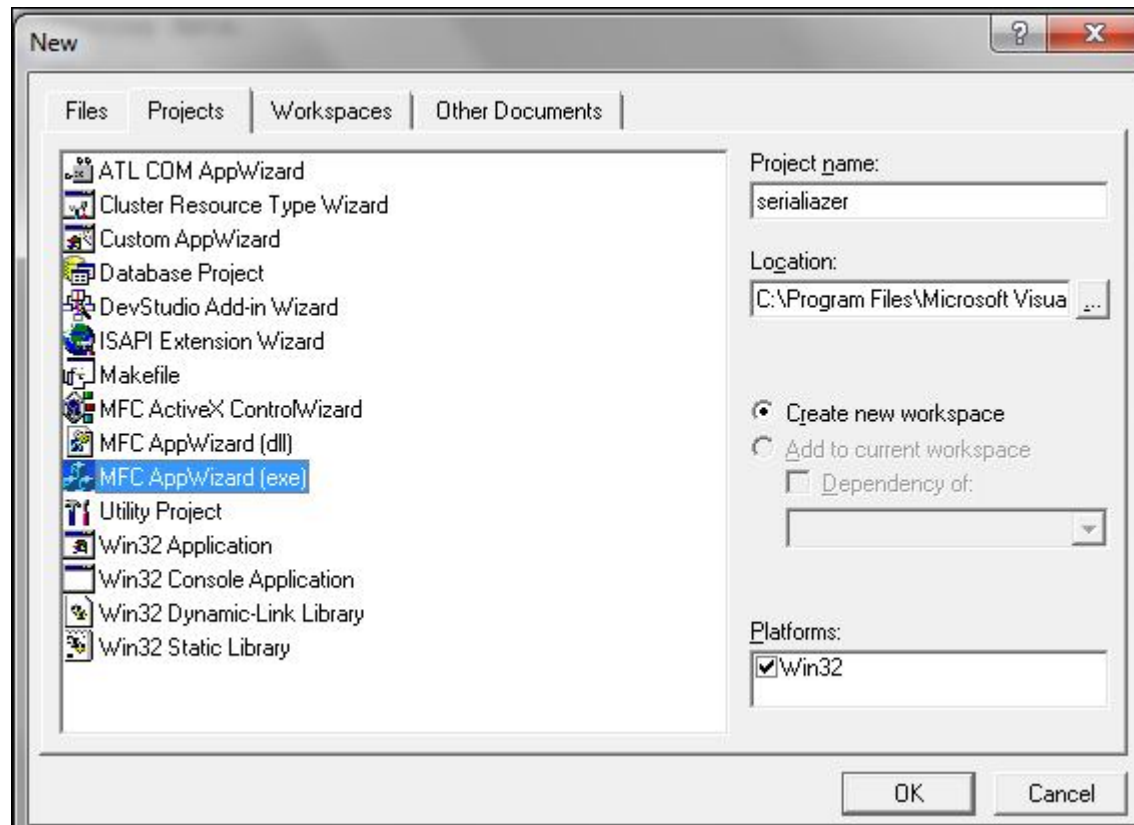


Create SDI application -> c++ header files -> ok

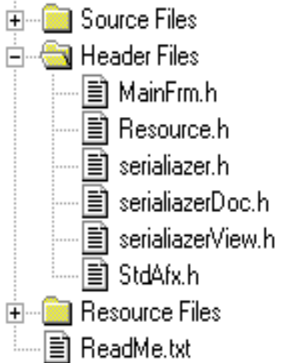
```
class CData{  
private:  
    CString data;  
public:  
    CData()  
    {  
        data = CString("");  
    }  
    void AddText(CString text)  
    {  
        data+=text;  
    }  
    void DrawText(CDC*pDC)  
    {  
        pDC->TextOut(0,0,data);  
    }  
    void ClearText()  
    {  
        data = "";  
    }  
};
```

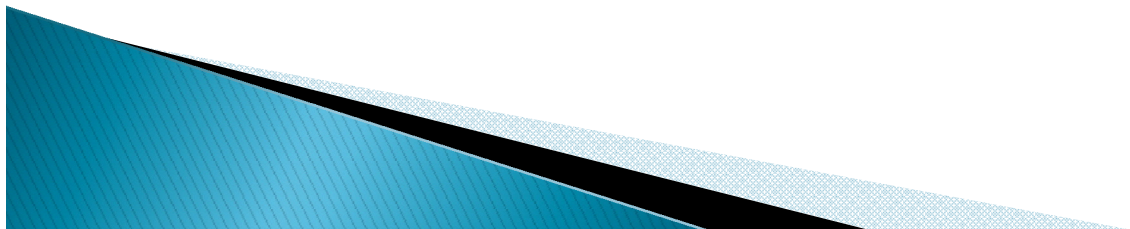


Create new SDI program named Serializer



serializerDoc.h

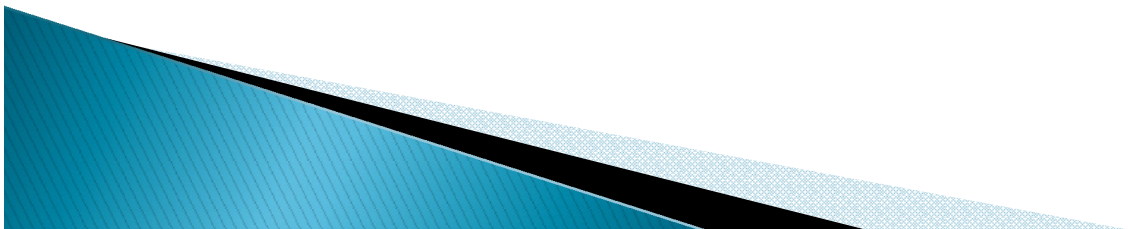
	<pre>#if !defined(AFX_SERIALIZERDOC_H__749F2629_CB4B_43BD_B901_D6FC8470DABE__INC #define AFX_SERIALIZERDOC_H__749F2629_CB4B_43BD_B901_D6FC8470DABE__INCLUDEI #if _MSC_VER > 1000 #pragma once #endif // _MSC_VER > 1000 #include "CData.h" class CSerializerDoc : public CDocument { protected: // create from serialization only CSerializerDoc(); DECLARE_DYNCREATE(CSerializerDoc) // Attributes public: CData DataObject; // Operations public:</pre>
---	--



SerialiazerView.cpp

```
void CSerialiazerView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: Add your message handler code here and/or call default

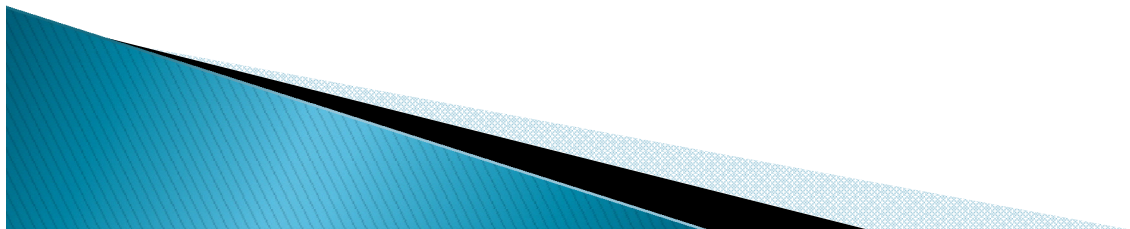
    CSerializerDoc*pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    pDoc->DataObject.AddText(CString(nChar));
    Invalidate();
    CView::OnChar(nChar, nRepCnt, nFlags);
}
```



SerializerView.cpp

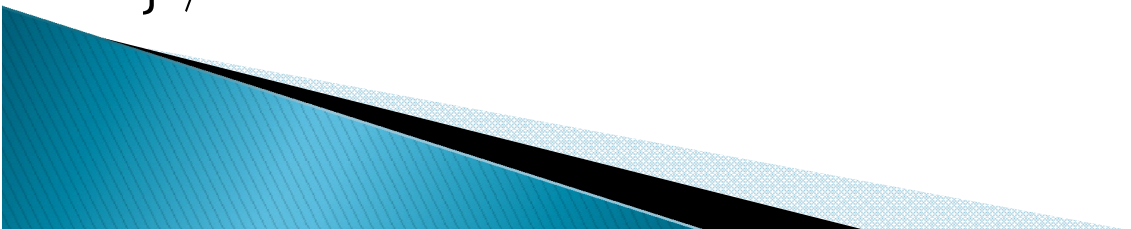
```
void CSerializerView::OnDraw(CDC* pDC)
{
    CSerializerDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    pDoc->DataObject.DrawText(pDC);
    // TODO: add draw code for native data here
}

////////////////////////////////////
// CSerializerView printing
```

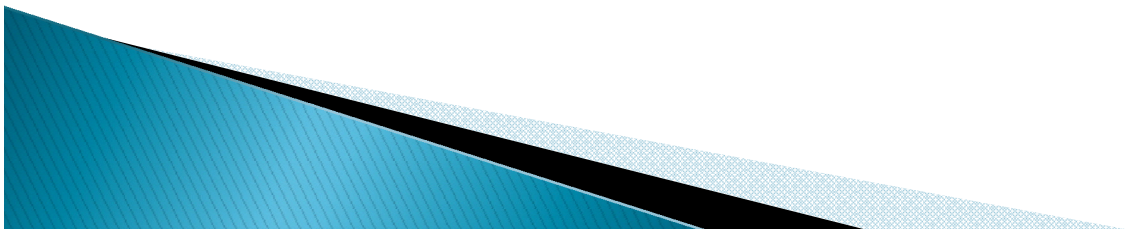


Adding Serialization to a class

```
class CData :public CObject
{
Private:
CString data;
DECLARE_SERIAL(CData);
Public:
CData(){data = CString("");}
void AddText(CString text){data += text;}
void DrawText(CDC* pDC){pDC->TextOut(0,0,data);}
void ClearText(){data="";}
void Serialize(CArchive& archive);
};
```



```
#include "stdafx.h"
#include "serializerDoc.h"
void CData: Serialize(CArchive& archive)
{
    CObject:Serialize(archive);
    if(archive.IsStoring()){archive << data;}
    else{archive >> data;}
}
```



```
void CSerializerDoc:: Serialize(CArchive& ar)
{
DataObject.Serialize(ar);
}
```

