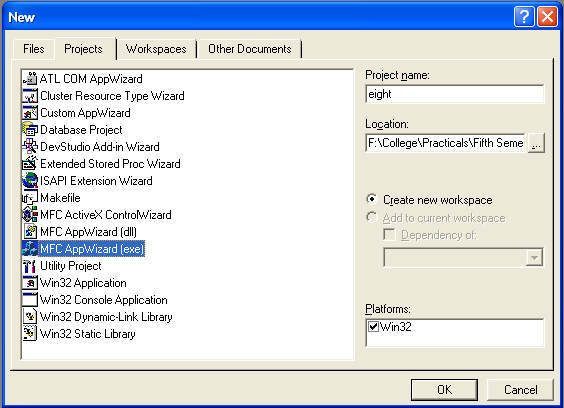
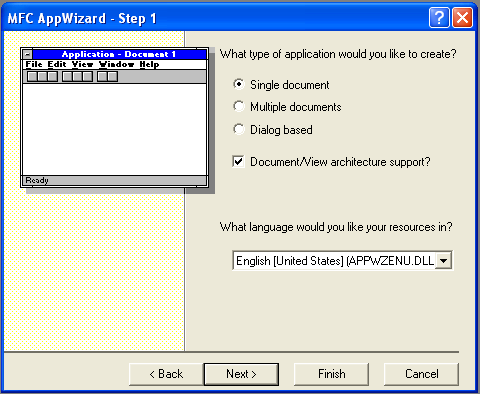
**STEPS**

1. Open Visual C++ and click the New item in the File menu and a new dialog box is opened
2. Now select “MFC AppWizard (exe)” entry
3. Give the new Project name in the Project name box and click OK. Clicking OK will start the Visual C++ AppWizard



1. We will accept all defaults except one- by default, AppWizard creates Multiple Documents program and we will change so that it creates a Single Document program



1. The AppWizard is asking for what database we want in our program; we will do “None” option selected. Keep pressing next until project information box is opened.
2. In the document header file, eightdoc.h:

class CEightDoc : public CDocument

{

protected: // create from serialization only

CEightDoc();

DECLARE\_DYNCREATE(CEightDoc)

* CString StringData;

.

.

.

.

}

1. Next we initialize the a object in the sevendoc.cpp file:

CeightDoc :: CeightDoc

{

* StringData="";

}

1. We initialize two variables x and y in view header file, eightView.h:

class CEightView : public CView

{

protected: // create from serialization only

CEightView();

DECLARE\_DYNCREATE(CEightView)

* CPoint CaretPosition;
* boolean CaretCreated;
* int x,y;

.

.

.

.

}

1. We then use Class Wizard to add the OnLButtonDown() and On\_Char() method to the program eight view class, CeightView. The two parameters nFlags indicates the state of various keys on the keyboard and the point parameter holds the mouse present location. The mouse location is stored in x and y variables

void CEightView::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

* x=point.x;
* y=point.y;
* CEightDoc \*pDoc=GetDocument();
* ASSERT\_VALID(pDoc);
* pDoc->StringData.Empty();
* Invalidate();

CView::OnLButtonDown(nFlags, point);

}

void CEightView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)

{

// TODO: Add your message handler code here and/or call default

CEightDoc \*pDoc=GetDocument();

ASSERT\_VALID(pDoc);

pDoc->StringData+=nChar;

Invalidate();

CView::OnChar(nChar, nRepCnt, nFlags);

}

1. Now open OnDraw() method from class tab in Ceightview to change the location of caret at (x,y) and to display caret at the end of the string data:

void CEightView::OnDraw(CDC\* pDC)

{

CEightDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if(!CaretCreated)

{

TEXTMETRIC textmetric;

pDC->GetTextMetrics(&textmetric);

CreateSolidCaret(textmetric.tmAveCharWidth/8,textmetric.tmHeight);

CaretPosition.x=CaretPosition.y=0;

SetCaretPos(CaretPosition);

ShowCaret();

CaretCreated=true;

}

pDC->TextOut(0,0,pDoc->StringData);

CSize size=pDC->GetTextExtent(pDoc->StringData);

HideCaret();

CaretPosition.x=size.cx;

SetCaretPos(CaretPosition);

ShowCaret();

// TODO: add draw code for native data here

}

1. Now run the program by Build Eight.exe and Execute Eight.exe item in the Build menu and type some text into it

**OUTPUT**

