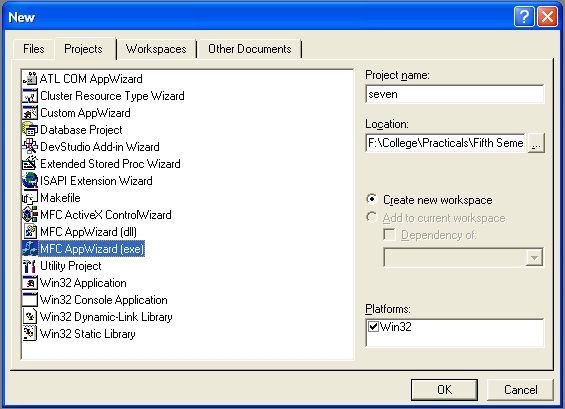
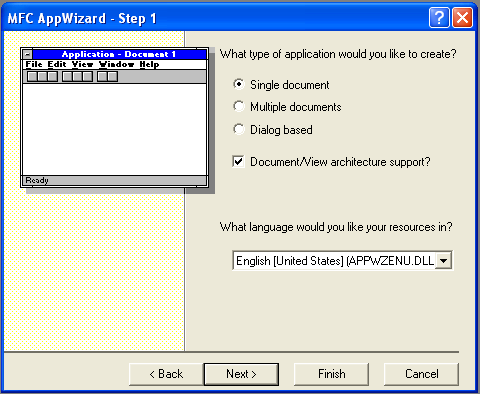
**STEPS**

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, sevendoc.h:

class CSevenDoc : public CDocument

{

protected: // create from serialization only

CSevenDoc();

DECLARE\_DYNCREATE(CSevenDoc)

* CString StringData;

.

.

.

.

}

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

CsevenDoc :: CsevenDoc

{

* StringData="";

}

1. We then use Class Wizard to add the OnChar() method to the program seven view class, CsevenView

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

{

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

* CSevenDoc \*pDoc=GetDocument();
* ASSERT\_VALID(pDoc);
* pDoc->StringData+=nChar;
* Invalidate();

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

}

1. Now we’ll create a Boolean variable named “CaretCreated” in the view object to keep track whether or not we’ve already created the caret.

class CSevenView : public CView

{

protected: // create from serialization only

CSevenView();

DECLARE\_DYNCREATE(CSevenView)

* boolean CaretCreated;

.

.

.

}

Now we use GetTextMetrics() method to determine the height and width of characters. We will make the caret as the height of our text, using textmetric.tmAveCharWidth/8.

1. Now open OnDraw() method from CSevenView.cpp

void CSevenView::OnDraw(CDC\* pDC)

{

CSevenDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if(!CaretCreated)

{

TEXTMETRIC textmetric;

pDC->GetTextMetrics(&textmetric);

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

}

.

.

.

.

}

We will store the caret position in a new CPoint object named CaretPosition. The CPoint class has two data members x and y which will hold the position of the caret

1. Now open sevenview.h:

class CSevenView : public CView

{

protected: // create from serialization only

CSevenView();

DECLARE\_DYNCREATE(CFifView)

* CPoint CaretPosition;

boolean CaretCreated;

.

.

.

.

}

1. Initially we set the caret’s position to (0,0) in OnDraw() method. Then caret position is set by SetCaretPos() method, to show the caret ShowCaret() method is used and set the Boolean flag CaretCreated to true
2. Next step is to move the caret as the user types the text and the caret also blinks. To place the caret at the end of the displayed text string, we use CSize object named size using GetTextExtent() method
3. To display caret at the end of the text string, we have to hide it by HideCaret() method
4. Finally after typed text the caret is visible.
5. Open OnDraw() method from the class tab in CSevenview:

void CSevenView::OnDraw(CDC\* pDC)

{

CSevenDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if(!CaretCreated)

{

* + 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 Seven.exe and Execute Seven.exe item in the Build menu and type some text into it

**OUTPUT**

