

# Chapter 12



## Windows Programming with the Microsoft Foundation Classes

# Elements of a Window (P.581)

- Let us go through them to be sure we have a common understanding of what the terms mean.
  - parent window, child window
  - border, size grip
  - title bar, title bar icon, status bar
    - system menu
      - click the title bar icon,
      - or right-click the title bar
  - client area
    - x increasing from left to right,
    - y increasing from top to bottom
  - minimize, maximize, close buttons

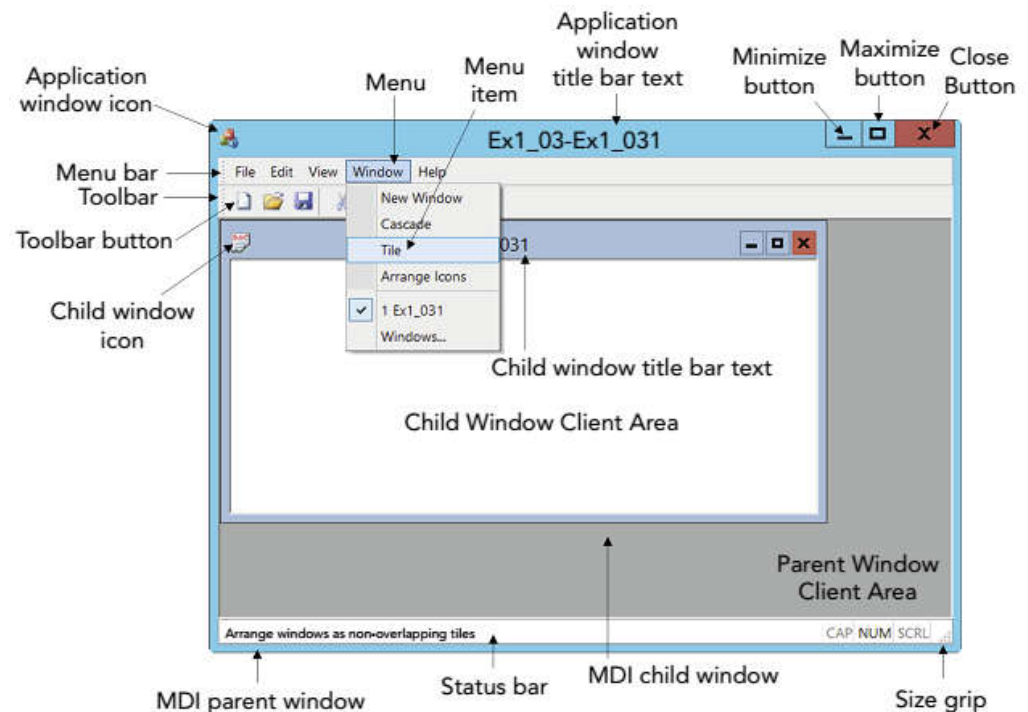


FIGURE 11-1

# The Microsoft Foundation Classes (P.605)

---

- MFC are a set of predefined classes.
  - These classes provides an **object-oriented approach** to Windows programming that encapsulates the Windows API.
    - Easy to use
    - Data Members & Member Functions
  - You will apply techniques you learned from the previous chapters, particularly those involving class **inheritance** and **virtual functions**.

# MFC Notation (P.605)

---

- All the classes in MFC have names beginning with **C**
  - CDocument
  - CView
- Data members of an MFC class are prefixed with **m\_**
  - m\_lpcmdLine
    - Explicitly showing the type of a variable in its name was important in the C environment, because of the lack of type checking
      - Hungarian notation (P.813, P.836)
    - However, C++ has strong type checking, so this kind of notation isn't essential, and will not be used heavily in this book.
    - However, the **p** prefix for pointers will be retained because this helps the code more readable.

# The Document/View Concept in MFC

---

- Document (P.614) – the collection of data
  - A document is not limited to **text**. It could be the data for a game, or the distribution of orange trees in California.
- View – how the data is to be displayed in a window, and how the user can interact with it.
  - A document object can have as many view objects associated with it as you want.

# A Document with Two Views (P.615)

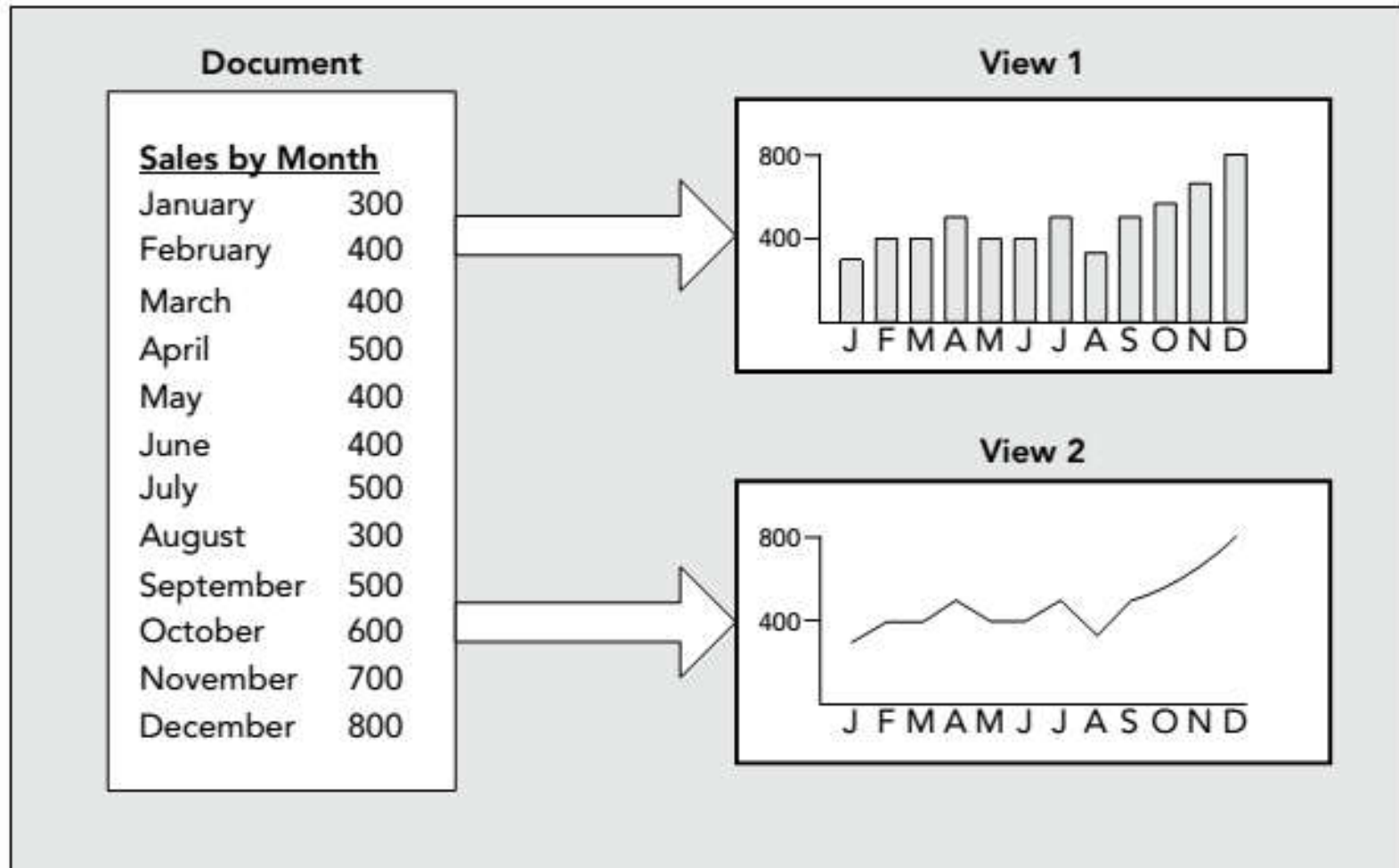


FIGURE 12-1

# Document Interfaces

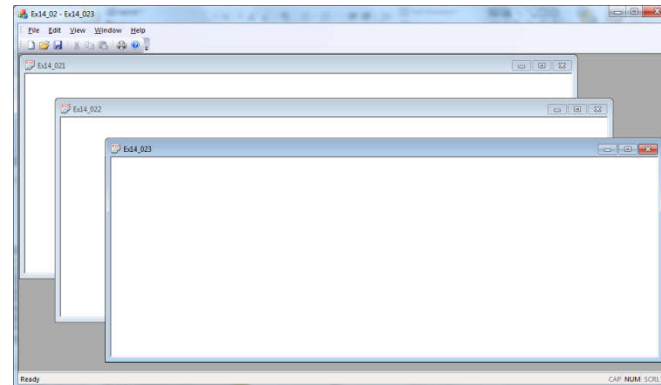
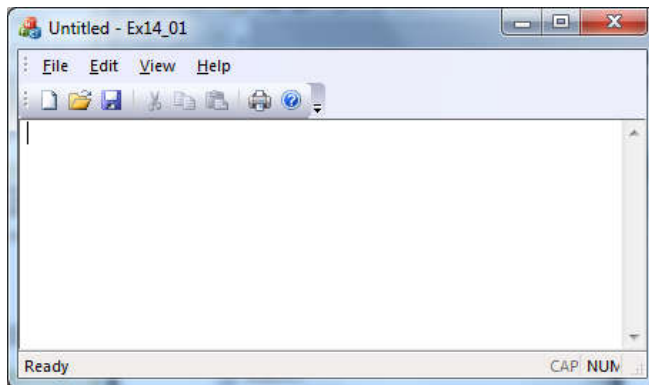
---

- SDI – Single Document Interface
  - Your application only open one document at a time.
- MDI – Multiple Document Interface.
  - Multiple documents can be opened in your application.
  - Each document is displayed in a child window of the application window.

# Document Template Classes

---

- MFC has two classes for defining document templates:
  - CSingleDocTemplate for SDI
  - CMultiDocTemplate for MDI





# Linking a Document and Its Views

---

- MFC incorporates a mechanism for integrating
  - a **document** with its **views**
    - a document object automatically maintains a list of pointers to its associated views
    - a view object has a pointer to the document
  - a **frame window** with a view
    - a frame window has a pointer to the currently active view object

# Document Templates (P.616)

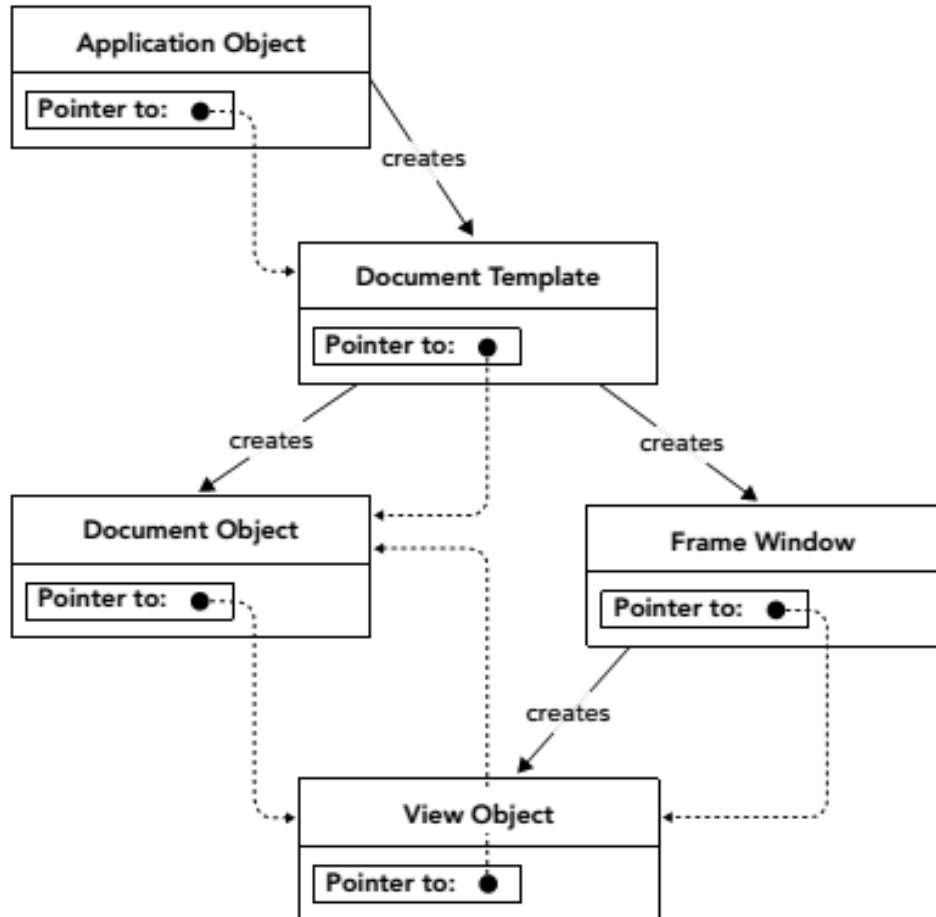


FIGURE 12-2

- A document template object creates document objects and frame window objects
  - If you have two or more documents of the same type, you need only one document template to manage them.
- View of a document are created by a frame window object.
- The application object creates the document template object.

# Your Application and MFC (P.617)

- Four basic classes that will appear in almost all your MFC-based Windows applications:
  - The application class
  - The document class
  - The view class
  - The frame window class

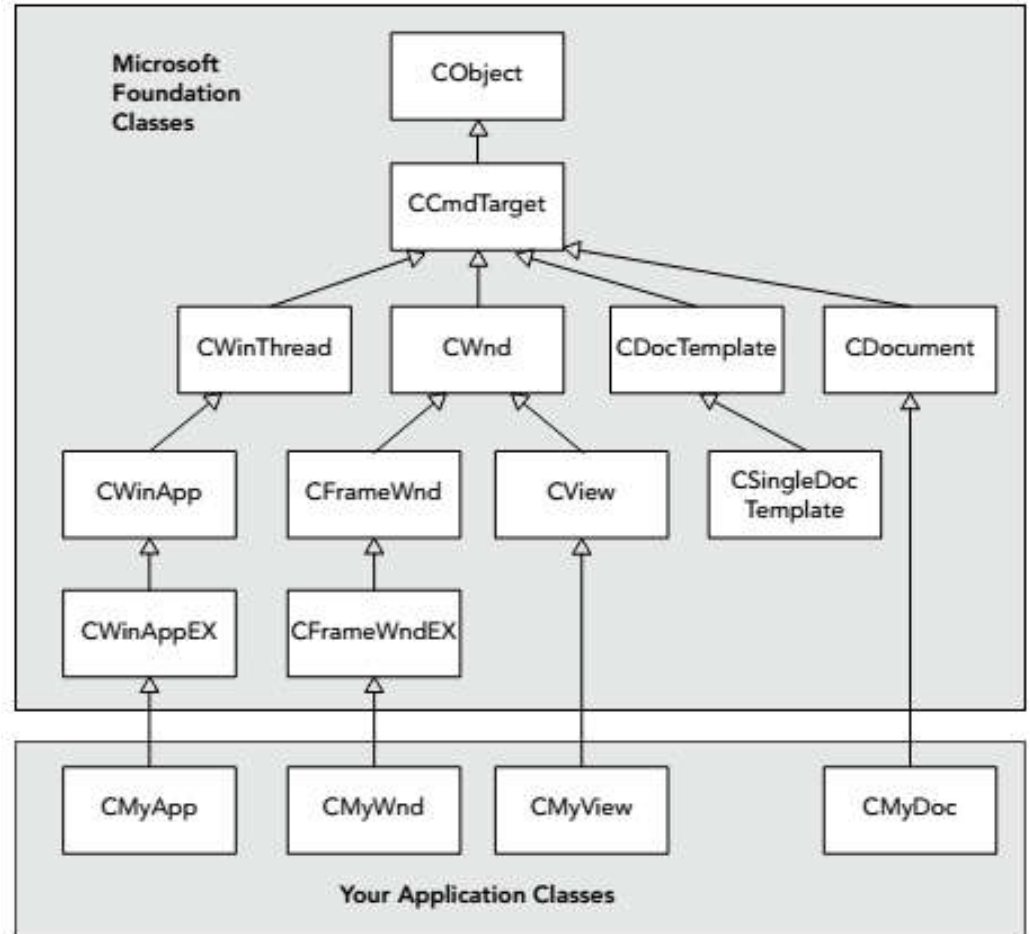


FIGURE 12-3

# Document / View Classes

---

- Your document class is derived from the CDocument class in the MFC library
  - You will add your own **data members** to store items that your application requires,
  - and **member functions** to support processing of that data.
- Your view class is derived from the CView class.
  - You define how data in your document will be displayed in a window.

# The Application Class

---

- ❑ The class CWinApp is fundamental to any Windows program written using MFC.
- ❑ An object of this class includes everything necessary for **starting**, initializing, running and **closing** the application.

```
class CMyApp: public CWinApp
{
    public:
        virtual BOOL InitInstance();
};
```

# The Window Class

---

- The CFrameWnd class provides everything for creating and managing a window for your application
  - All you need to add to the derived window class is a **constructor**.

```
class CMyWnd: public CFrameWnd
{
public:
    // Constructor
    CMyWnd()
    {
        Create(0, L"Our Dumb MFC Application");
    }
};
```