

The Multiple Document Interface (MDI)

Unit 4

Multiple Document Interface (MDI)

- A **The Multiple Document Interface (MDI)** if the user can open more than one document in the application without closing it.
- The application provides a parent frame that acts as the main frame of the computer program, inside this it allows another frame child frame
- Using MDI application, a user can create a document, open another while the first is still displaying and even create new documents or open additional ones.

Features Of MDI

- If one document is maximized all the document are maximized.
- If no document is maximized, each document can be in two states: minimized or restored.
- The child document will be on main frame. If a document get minimized it could display inside the main frame.
- If the document is not minimized it will be on the main frame as original size or user defined size.
- To differentiation the parent and children, the application uaes 2 menus: 1) main frame 2) child for each document
- Both menus uses the same menu bar.

When MDI application is created, the classes supported are:

- The CWinApp derived class
- The CMDIFrameWnd derived class
- The CMDIChildWnd derived class
- The CDocument derived classe
- The CView Derived class

2 MDI Derived classes which are different from SDI that is

- The CMDIFrameWnd derived class
- The CMDIChildWnd derived class

The CMDIFrameWnd

- Its derived from CMainFrame, is the main frame of the application, in this frame menus and tools area attached. It provides an enclosed space on the desktop

- **The CMDIChildWnd**

Its derived from CChildFrame class, this frame that holds the CView class

It passes the message and events to the view class for processing or display.

Using MFC AppWizard 2 kinds of document /view architectures :

SDI : only one document is displayed in a single view, which contains in the main frame window (Cframewnd)

When file → New command is selected, the document data are destroyed and replaced by a new content. There is no file → close command

The document view frame are destroyed together. When user exits the application

This is managed by CSingleDocTemplate object

MDI

- There can be zero, one or several documents.
- Each document is displayed in a separate view, which contained in a child frame window (CMDIChildWnd)
- All child frame windows are contained in the main window (CMDIFrameWnd)
- When file → new a new document is opened and new view and a new child frame are created.
- When file → close the active document, the active view and active child frame is destroyed.
- **This is managed by CMultiDocTemplate object**

MDI advantages

- 1) MDI a single menu bar and tool bar is shared btw all child windows.
- 2) An application child windows can be hidden/minimized/maximized as a whole
- 3) Faster and more memory efficient since the application are been shared and only the document changes
- 4) The speed of switching btw the internal windows being faster then having OS switch btw external window.
- 5) Application have keyboard shortcuts to quickly jump to the functionality needed

MDI disadvantages

- 1) MDI can make it more difficult to work with several application at once by restricting the ways in which they have arranged together
- 2) The shared menu might change, can then confusion to some users
- 3) MDI child windows behave differently from those in single document interface applications, it requiring user to learn two different windowing concepts.

Splitter Windows

- A splitter window appears as a special type of frame window that holds several views in panes.
- A windows might contain normal views or one window might contain page layout view and another contains outline view.
- Each windows will have scroll bar.
- the user can use the splitter window to display multiple views.
- the object class `CsplitterWnd` represents the splitter window

The application of splitter windows are

- 1) Split the window on creation
- 2) Split the window by choosing a menu command
- 3) By dragging a splitter box on the windows scroll bar
- 4) After the window is split, the user can move the splitter bars with the mouse to adjust the panes or resize
- 5) Splitter window can be used in both SDI and MDI application

2 types of splitter windows are:

- Static
- Dynamic

Exception Handling

- While debugging a program a user normally encounters two types of errors :
 - 1) Logical error and
 - 2) Syntactical error

VC++ supports three kinds of exception handling:

- 1) C++ exception handling
- 2) Structure exception handling (SEH)
- 3) MFC exceptions

C++ exception handling

- Try, throw and catch statements implement exception handling, with C++ execution handling.
- A program can communicate unexpected events to a higher execution context that is better able to recover from such abnormal events.
- TRY: a try block identifies a block of statements, for which a particular exception will be activated. The try block is indicated by TRY, followed by a block of statement enclosed within braces

- Throw: on detecting an exception, its thrown using throw sts in the try block.
- Catch: it indicates, catching the exception.

A catch block catches the exception thrown by the throw sts in the try block and handles it

A catch block begins with the keyword catch following by a type declaration within parentheses, indicating the type of execution to which it responds.

Try

{

Throw Exception;

//sts to detect and throw an exception

}

Catch(datatype arg)

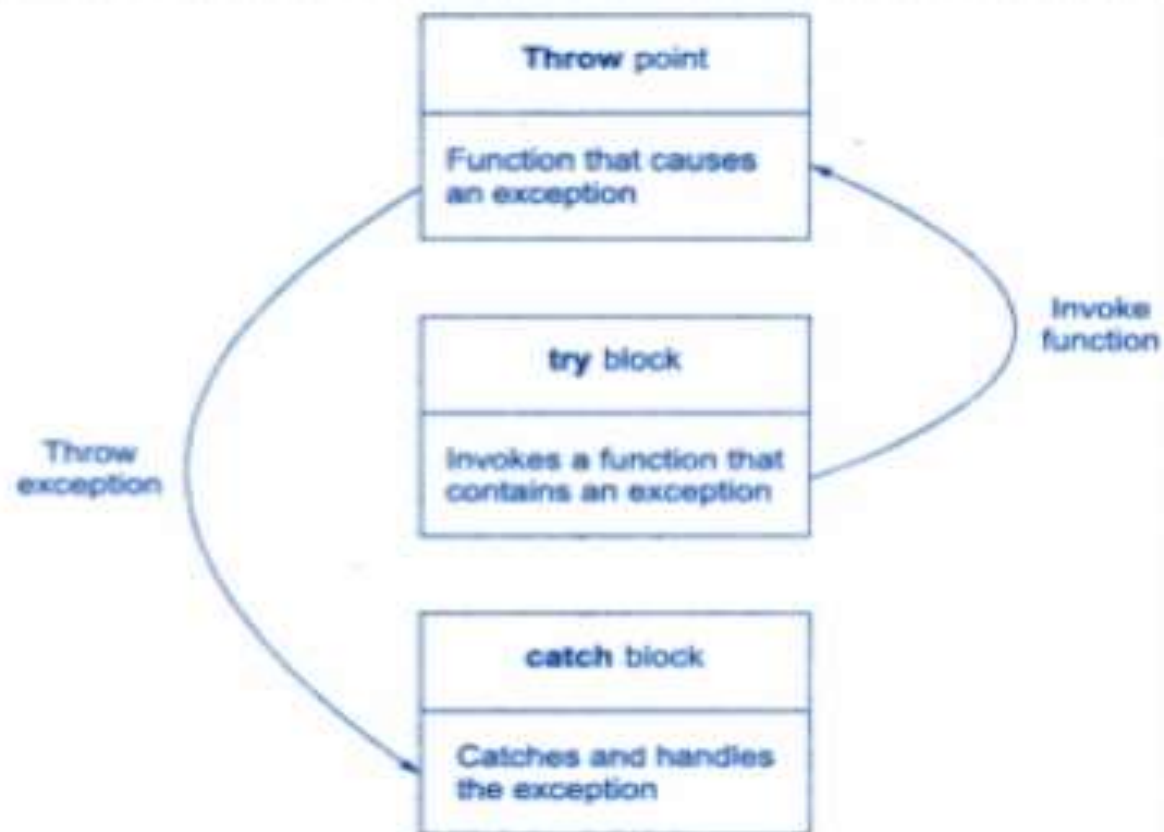
{

// sts to handle the exception

}

Exceptions thrown by functions

- Mostly exceptions are thrown by functions that are invoked from within the try blocks.
- The point at which the throw is executed is called the throw point.



Throwing Mechanism

- An exception is detected by throw statement. A throw statement can have one of the following forms:

1) Throw

2) Throw Exception;

3) Throw (exception);

A throw expression without argument is used to rethrow an exception. This point at which throw is executed is known as the **throw point**.

Multiple catch statements

```
Try
{
//sts
}
Catch (type1 arg1)
{
// sts to catch arg1 of type1
}
---
---
Catch (typeN argN)
{
// sts to catch argN of typeN
}
```

Its possible to have multiple catch handlers associated with each try block.

Its similar to a switch sts.

Structure exception handling (SEH)

- SEH can be used for a powerful and a more readable way to handle errors.

```
Try
{
//code that is expected to raise an exception
}
Catch (Exception ex)
{
// code that can handle an error
}
__finally
{
// code to do any final cleanup
}
```

MFC Exceptions

program already use the macros, so c++ exception can also be used.

During pre-processing, the macros evaluate to the exception handling keywords defined in the VC++ implementation of the C++ language.

OLE (object linking and embedding)

- Its Microsoft component document
- With OLE files and applications can be dynamically linked together
- OLE allows the user to integrate data from different application
- Object linking allows users to share a single source of data for a particular object.

OLE Features

- Objects
- Files
- Data
- Embedding

Benefits of OLE

- Is to display visualizations data from other programs that the host program is not normally able to generate itself.
- Its primary use is for managing compound documents
- Its also used for transferring data btw different application using drag and drop operation
- The embedding uses of multimedia in webpages, video embed, TML and XHTML

Dynamic Link Libraries (DLLs)

(It contains functions, classes and resources)

- It contains pre-defined functions, which are linked to an application program when it is loaded dynamically.
- It provides a powerful and flexible graphics user interface to the OS
- Libraries of functions, save the programmers, the task of re-creating new procedures for a common operation likes character font changing of justifying text.. Etc
- Dlls include KERNEL32.dll,USER32.dll,KEYBOARD, SYSTEM.DRV, MOUSE.DRV

DLL libraries can be linked to execute in 2 ways

- Static linking and
- Dynamic linking

DLL uses 2 kinds of function

- 1) Exported
- 2) Internal

1. Object oriented programming

Class: is a portion of the code that defines the properties methods and events.

Class is an abstract representation that enclose the characteristics of one/more objects that will be created during execution

A class is a template or blueprint for an object.

Object: Instance of a class is called object.

an object is unit of software that contains a collection of related methods and data

An object contains the characteristics of that class

Data Abstraction: denotes the essential characteristics of an object that distinguish it, from all other objects and provide crisply defined conceptual boundaries

Data Encapsulation: is wrapping of data and functions into a class. It helps to maintain data security of the class members.

Inheritance: it's the process by which an object inherits the properties of another object belonging to another class.

Polymorphism: defines as the ability of the related object to respond to the same message with different but appropriate actions.

2. Class modules

- Its used to implement or define objects
- It represent a logical object
- Its useful while creating any kinds of ActiveX code component, since they represent a logical object in the project
- Goto project menu → add class module → change the name property to Cstudent → write the code

Private m_name As string

Private m_rno As string

Private m_tmarks As Integer

- Class modules are used to define a class. It serves as a template for creating objects.
- The class model data exists separately for each instance of the class.
- It exists for the life of the program while class module data exist for the lifetime of object. When the object is created and destroyed

Creating Properties

- Can define a property for object in two ways:

Define public variables

Create public property procedure within class module

Using public variables:

```
Public cardno AS integer
```

```
Dim Demo As class1 // creates an instance of class1
```

```
Set Demo = New Class1
```

```
Demo cardno = 1250
```

Class methods: are sub procedure or function procedure

```
Public sub showgrandemark()
```

```
MsgBox m_totmarks
```

```
End sub
```


Class module Events

- 1) Initialize event: occurs when an instance of a class is created, but before any properties have been set.

initialize event is used to initialize any data used by the class

```
Private sub class-initialize()
```

```
Cardno = 1234
```

```
End sub
```

2) The terminate event: occurs when the object goes out of scope or is set nothing.

This event is used to save information, unload forms

```
Private sub class-terminate()
```

```
End sub
```

3. Windows API

- It provides a set of routines, functions , messages and structures, which allow programmers to build application running on window's platform. This is known as windows application interface (API)
- Basic services ----- kernel32.dll, advapi32.dll(advanced services)
- Graphics Device Interface ----- win32k.sys
- User interface----- comdlg32.dll
- Multi media----- winmm.dll
- Network services---- netBIOS, RPC
- System services– access resource features of OS

1. File Handling

- Many program uses files. I,e a program saves data into database or just keeps information for its own use. We need to have files
- These files can be stored in secondary storage devices I,e disk drives, magnetic disks and drives and optical disk

3 main ways to access file in VB are :

- 1) Sequential files
- 2) Random access files
- 3) Binary files.

Sequential Files

- The files are like tape cassettes.
- Data is read in a sequential order.
- This file can read and write in order from beginning to end
- To use the file, entire file is processed from the start to end
- Its useful when storing small amount of data, where access speed is not an issue.
- Sequential files are often organized into text strings in VB
- It can be even used for large amount of data to be stored, provided it all has to be processed at one time.

Random File

- Random file is a file where all records are accessible individually.
- Its like a CD where you can jump to any track.
- Its useful when there is a large quantity of data to store and it has to be available quickly.
- The user should know the part of the data to access from the large data, bcoz it doesn't have time to search via 10,000 records individually to locate the correct one.
- Therefore this method of storage becomes popular when hard-disk drives were developed.

Binary file

- Compacted from random file.
- Data is stored at the byte level and can read, write individual bytes to the file.
- This makes the file access very fast and efficient.

2. ACCESS MODE

- **There are 5 access modes in the file**

- 1) **Input:** open for sequential input, the file will be read sequentially starting at the beginning.
- 2) **Output:** open for sequential output, records will be written sequentially starting at the beginning, if the file does not exist it is created if it does exist its overwritten
- 3) **Random:** open for random read and write, any specific record can be accessed.
- 4) **Append:** sequential output added to the end of an existing file, if file does not exist its created. It does not overwrite the file.
- 5) **Binary:** open for binary read and write access is at byte level.

Open and close files

- To work with files the first command to the file is to open Command
- Open assigns the file to a numbered file handle or channel or buffer

Syntax:

Open "filename" [file mode] As # filename

Eg: open app.path &"\myFile.dat" for output As #1

Myfile.dat is the name of the file.

App.path is the path to the currently open file

For output means that the file is opened for writing the records sequential

AS #1 is the file name, is assigned a file handle #1, #2 so on

Syntax:

Close #filename1 [,#filename1, #filename2...]

Close #1, #2, #3

It closes all the open files.

random access file handling with read and write process

- The command to write records to random file is PUT
- Format:

Put #Filename, [RecordNumber], Variable

Where,

filename is the number of file to write, the record number begins with 1, variable is used to read write any data type.

- The command to read records to random file is GET
- Format:

GET #Filename, [RecordNumber], Variable

Where,

filename is the number of file to read, the record number is the number of the record to read and variable is used to read write any data type.

3. File system Object (FSO)

- FSO model provides an object-based tool to work with folders and files.
- FSO facilitate the programmers to manipulate files, directories and drives.
- They also allows the user to read and write text to sequential files.
- Project menu → References → select Microsoft scripting runtime

- The FSO provides a number of properties and methods for manipulating file system and for reading and writing text files.

- **Properties are:**

Atendof line – Boolean, specifies end of the file.

Atendofstream – Boolean specifies end of the stream

Column – long specifies current character position

Line – Long specifies current line number

- **Methods:**

Close – close the text stream

Read – read a specific number of character

Read all , readline, skipline, write, writeline

Designing Help file

- Help file which can take many forms. Its an external document providing interactive help for searching for some specific topics
- Help support can be added to a VB application

Types of help files:

- 1) **WinHelp** – its traditional windows with an .hlp extension
- 2) **HTML Help** – is based on web technology and use .chm help file
- 3) **Context-Sensitive help** – is an application saves the users time trying to find information . This type of help gets the required information quickly and directly by being “aware” of which has focus, when the user presses F1