

New
them

Class overview

**(categorizes and describes the classes in the
Microsoft Foundation Class Library (MFC))**

Microsoft Windows was designed long before the C++ language became popular. Because thousands of applications use the C-language Windows application programming interface (API), that interface will be maintained for the foreseeable future.

Any C++ Windows interface must therefore be built on top of the procedural C-language API. This guarantees that C++ applications will be able to coexist with C applications.

The Microsoft Foundation Class (MFC) Library is an object-oriented interface to Windows that meets the following design goals:

- *Significant reduction in the effort to write an application for Windows.*
- *Execution speed comparable to that of the C-language API.*
- *Minimum code size overhead.*
- *Ability to call any Windows C function directly.*
- *Easier conversion of existing C applications to C++.*
- *Ability to leverage from the existing base of C-language Windows programming experience.*
- *Easier use of the Windows API with C++ than with C.*
- *Easier to use yet powerful abstractions of complicated features such as ActiveX controls, database support, printing, toolbars, and status bars.*
- *True Windows API for C++ that effectively uses C++ language features.*

The overview categorizes and describes the classes in the Microsoft Foundation Class Library (MFC) version 7.0. The classes in MFC, taken together, constitute an application framework — the framework of an application written for the Windows API. Your programming task is to fill in the code that is specific to your application.

The library's classes are presented here in the following categories:

- [Root Class: CObject](#)
- [MFC Application Architecture Classes](#)
 - [Application and Thread Support Classes](#)
 - [Command Routing Classes](#)
 - [Document Classes](#)
 - [View Classes \(Architecture\)](#)
 - [Frame Window Classes \(Architecture\)](#)
 - [Document-Template Classes](#)
- [Window, Dialog, and Control Classes](#)
- [Drawing and Printing Classes](#)
 - [Output \(Device Context\) Classes](#)
 - [Drawing Tool Classes](#)
- [Simple Data Type Classes](#)
- [File and Database Classes](#)
 - [File I/O Classes](#)
 - [ODBC Classes](#)
 - [OLE DB Classes](#)
- [Internet and Networking Classes](#)
 - [ISAPI Classes](#)
 - [Windows Sockets Classes](#)
 - [Win32 Internet Classes](#)
- [OLE Classes](#)
 - [OLE Container Classes](#)
 - [OLE Server Classes](#)
 - [OLE Drag-and-Drop and Data Transfer Classes](#)
 - [OLE Automation Classes](#)



Most of the classes in the Microsoft Foundation Class (MFC) Library are derived from a single base class at the root of the class hierarchy. **CObject** provides a number of useful capabilities to all classes derived from it, with very low overhead. For more information about **CObject** and its capabilities,

Application Classes supply functionality common to most applications.

Typically, you do so by deriving new classes from the architecture classes, and then adding new members or overriding existing member functions.

Application wizards generate several types of applications:

SDI (single document interface);

MDI (multiple document interface) applications that use the document/view architecture.

Document/view applications contain one or more sets of documents, views, and frame windows.

All MFC applications have at least two objects: an application object derived from **CWinApp**, and some sort of main window object, derived (often indirectly) from **CWnd**.

Applications that use document/view architecture contain additional objects. The principal objects are:

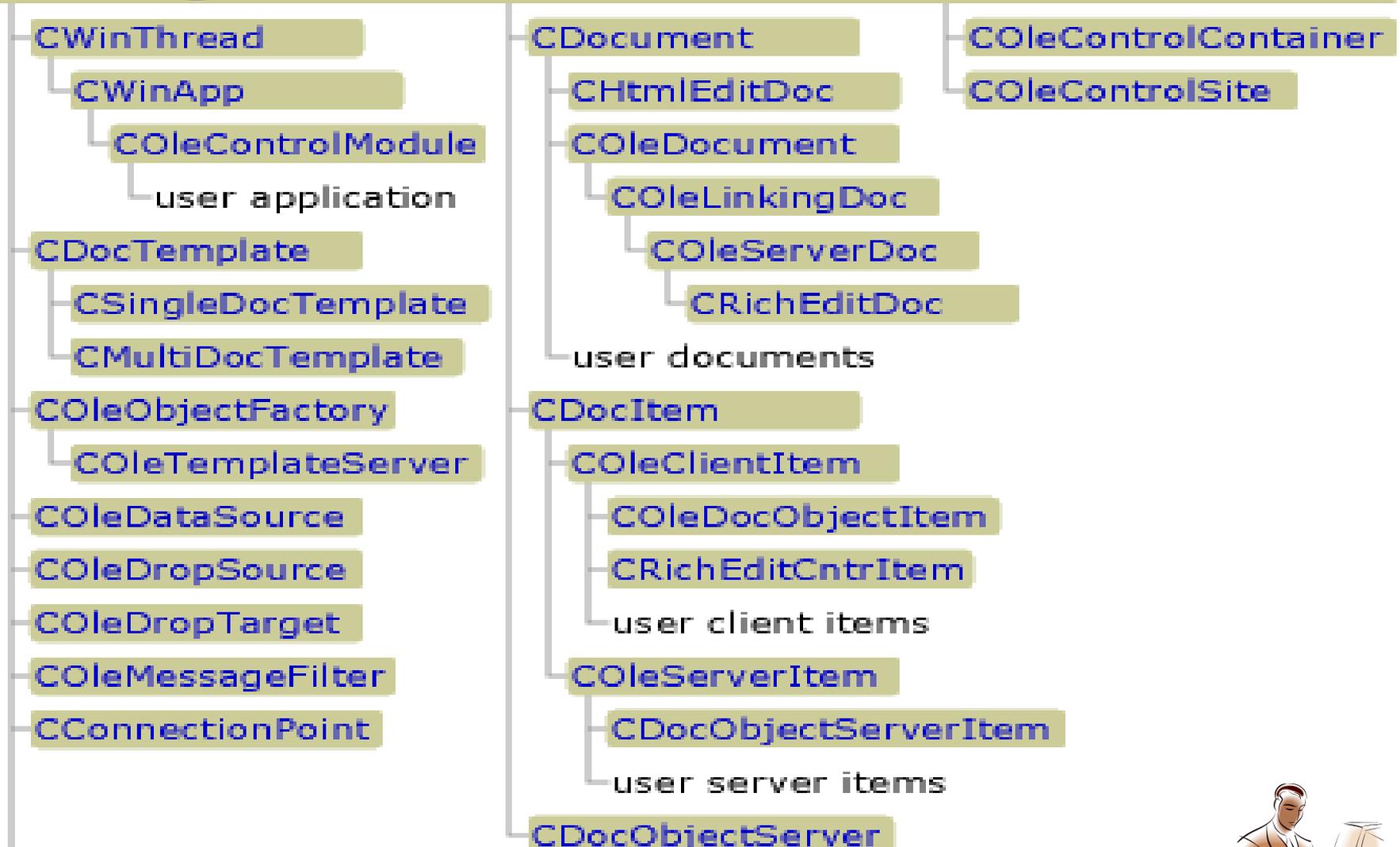
- An application object derived from class **CWinApp**, as mentioned before.
- One or more document class objects derived from class **CDocument**. Document class objects are responsible for the internal representation of the data manipulated in the view. They may be associated with a data file.
- One or more view objects derived from class **CView**. Each view is a window that is attached to a document and associated with a frame window. Views display and manipulate the data contained in a document class object.

Document/view applications also contain frame windows (derived from **CFrameWnd**).

COject

Application Architecture

CCmdTarget



user objects

Exceptions

CException

CArchiveException

CDaoException

CDBException

CFileException

CInternetException

COleException

COleDispatchException

CSimpleException

CMemoryException

CNotSupportedException

CResourceException

CUserException

File Services

CFile

CMemFile

CSharedFile

COleStreamFile

CMonikerFile

CAsyncMonikerFile

CDataPathProperty

CCachedDataPathProperty

CSocketFile

CStdioFile

CInternetFile

CGopherFile

CHttpFile

Graphical Drawing

CDC

CClientDC

CMetaFileDC

CPaintDC

CWindowDC

Control Support

CDockState

CImageList

Graphical Drawing Objects

CGdiObject

CBitmap

CBrush

CFont

CPalette

CPen

CRgn

CWnd

Frame Windows

- CFrameWnd
 - CMDIChildWnd
 - user MDI windows
 - CMDIFrameWnd
 - user MDI workspaces
 - CMiniFrameWnd
 - user SDI windows
 - COleIPFrameWnd

CSplitterWnd

Control Bars

- CControlBar
 - CDialogBar
 - COleResizeBar
 - CReBar
 - CStatusBar
 - CToolBar

Property Sheets

- CPropertySheet

Dialog Boxes

- CDialog
 - CCommonDialog
 - CColorDialog
 - CFileDialog
 - CFindReplaceDialog
 - CFontDialog
 - COleDialog
 - COleBusyDialog
 - COleChangeIconDialog
 - COleChangeSourceDialog
 - COleConvertDialog
 - COleInsertDialog
 - COleLinksDialog
 - COleUpdateDialog
 - COlePasteSpecialDialog
 - COlePropertiesDialog
 - CPageSetupDialog
 - CPrintDialog
 - CPrintDialogEx
 - COlePropertyPage
 - CPropertyPage
 - user dialog boxes
 - CDHtmlDialog
 - CMultiPageDHtmlDialog

Views

- CView
 - CCtrlView
 - CEditView
 - CListView
 - CRichEditView
 - CTreeView
 - CScrollView
 - user scroll views
 - CFormView
 - user form views
 - CDaoRecordView
 - CHtmlEditView
 - CHtmlView
 - COleDBRecordView
 - CRecordView
 - user record views

Controls

- CAnimateCtrl
- CButton
 - CBitmapButton
- CComboBox
 - CComboBoxEx
- CDateTimeCtrl
- CEdit
- CHeaderCtrl
- CHtmlEditCtrlBase
 - CHtmlEditCtrl
 - CHotKeyCtrl
 - CIPAddressCtrl
 - CLinkCtrl
 - CListBox
 - CCheckListBox
 - CDragListBox
 - CListCtrl
 - CMonthCalCtrl
 - COleControl
 - CProgressCtrl
 - CReBarCtrl
 - CRichEditCtrl
 - CScrollBar
 - CSliderCtrl
 - CSpinButtonCtrl
 - CStatic
 - CStatusBarCtrl
 - CTabCtrl
 - CToolBarCtrl
 - CToolTipCtrl
 - CTreeCtrl



Graphical Drawing

CDC

CClientDC

CMetaFileDC

CPaintDC

CWindowDC

Control Support

CDockState

CImageList

Graphical Drawing Objects

CGdiObject

CBitmap

CBrush

CFont

CPalette

CPen

CRgn

Menus

CMenu

Command Line

CCommandLineInfo

ODBC Database Support

CDatabase

CRecordset

user recordsets

CLongBinary

DAO Database Support

CDaoDatabase

CDaoQueryDef

CDaoRecordset

CDaoTableDef

CDaoWorkspace

Synchronization

CSyncObject

CCriticalSection

CEvent

CMutex

CSemaphore

Arrays

CArray (template)

CByteArray

CDWordArray

CObArray

CPtrArray

CStringArray

CUIIntArray

CWordArray

arrays of user types

Lists

CList (template)

CPtrList

CObList

CStringList

lists of user types

Maps

CMap (template)

CMapWordToPtr

CMapPtrToWord

CMapPtrToPtr

CMapWordToOb

CMapStringToPtr

CMapStringToOb

CMapStringToString

maps of user types

Internet Services

CInternetSession

CInternetConnection

CFtpConnection

CGopherConnection

CHttpConnection

CFileFind

CFtpFileFind

CGopherFileFind

CGopherLocator

Classes Not Derived

Internet Server API

CHttpArgList

CHtmlStream

CHttpFilter

CHttpFilterContext

CHttpServer

CHttpServerContext

Run-Time Object Model Support

CArchive

CDumpContext

CRuntimeClass

Structures

CCreateContext

CHttpArg

CMemoryState

COleSafeArray

CPrintInfo

Simple Value Types

CFileTime

CFileTimeSpan

CPoint

CRect

CSize

CSimpleStringT

CStringT

CFixedStringT

CTime

CTimeSpan



