

An Introduction to Visual C++ and MFC for Visual Basic Developers

by

Richard Hale Shaw

Richard Hale Shaw is a Contributing Editor to PC Magazine, Windows Tech Journal, MFC Journal, Visual Basic Programmer's Journal and Microsoft Systems Journal. He's also the editor of NT Developer Journal and writing *Visual Programming++* for Addison-Wesley. He can be reached at MCIMAIL 399-8368 or CompuServe 72241,155.

In this Session:

Why VC++?

An Introduction to MFC

Let's Build an application with VC++:

- Development environment**

- AppWizard**

- AppStudio**

- ClassWizard**

- Etc.**

Why Visual C++?

Copyright @ Richard Hale
Shaw

It's C++:

- Portable, Fast, Small

- Easy to build highly modular applications

- Easy to build flexible, *reusable* code

Produces native code executables

- No interpreted code

- Small programs (minimum 20k with MFC DLLs)

- Fast programs

MFC:object-oriented application support

Visual Development *a la* VB -- and more

- Wizards to generate/modify code

- Full support for VBX 1.0 controls

Why Visual C++?

Copyright @ Richard Hale
Shaw

Portable to Win16/Win32

Built-in support for building:

- OLE object containers

- OLE object servers

Built-in OLE Automation Server

- Perfect companion to VB (Automation-Client)

Plus:

- ODBC and more

Introducing the Microsoft Foundation Class Library (MFC)

What's an Application Framework?

Why Use MFC?

Features of MFC

MFC Message Mapping

MFC's Documents and Views

What's an Application Framework?

'Super' class library

Incorporates all essential features of an application

Provides a structure that you can build onto

Encapsulates standard application operations

In the case of MFC, encapsulates most standard Windows operations and more

Simplifying Windows Development

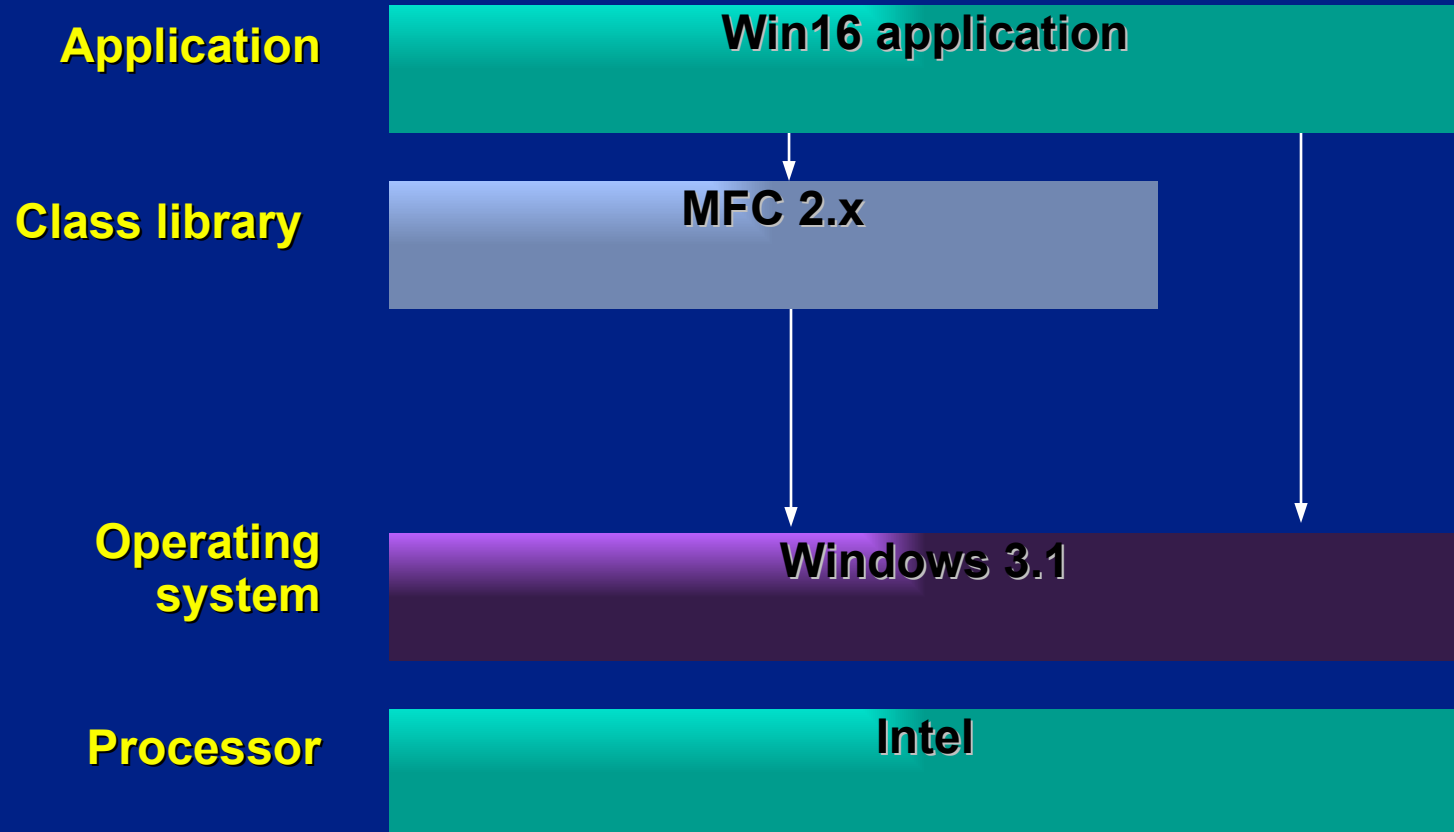
Application Frameworks:

- Provide a base for building Windows applications
- Encapsulate a great deal of the routine work
- Provide abstraction layers to make it easier, simpler
- Use polymorphism so that one interface serves multiple purposes

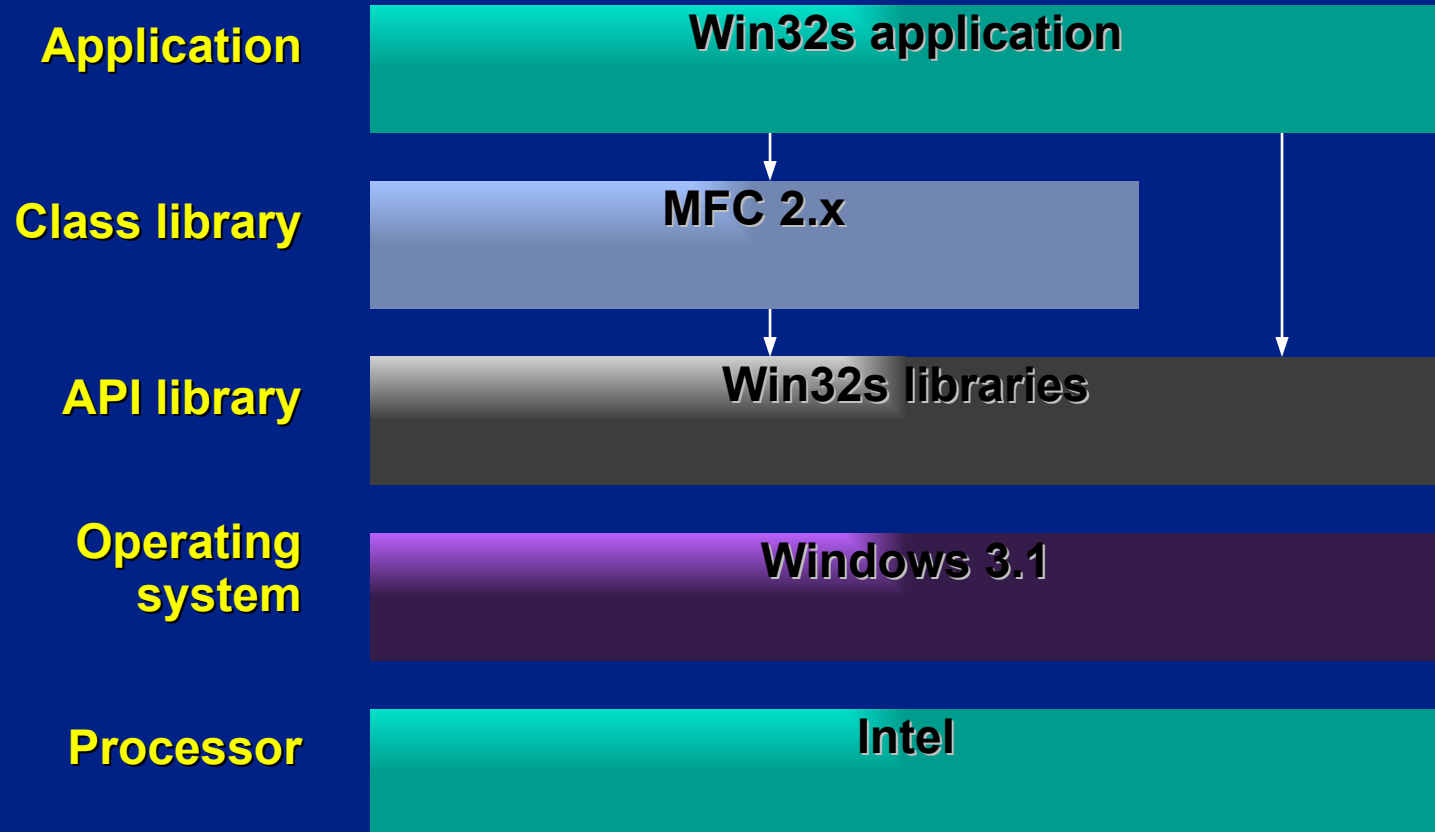
Visual C++ and the Microsoft Foundation Class Library (MFC) application framework

- Make application framework programming under Windows even easier using *visual* development techniques.

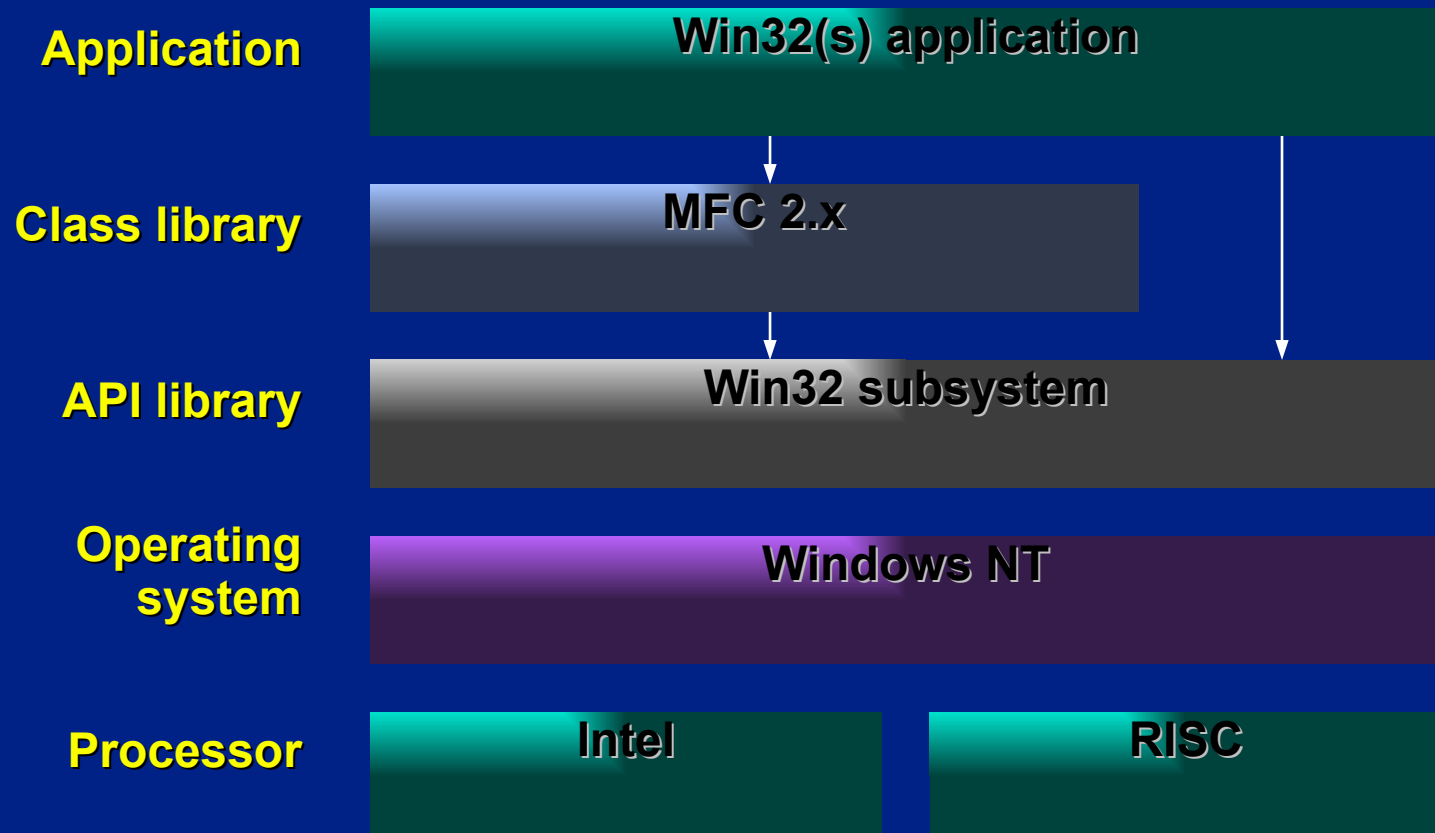
Win16 and MFC



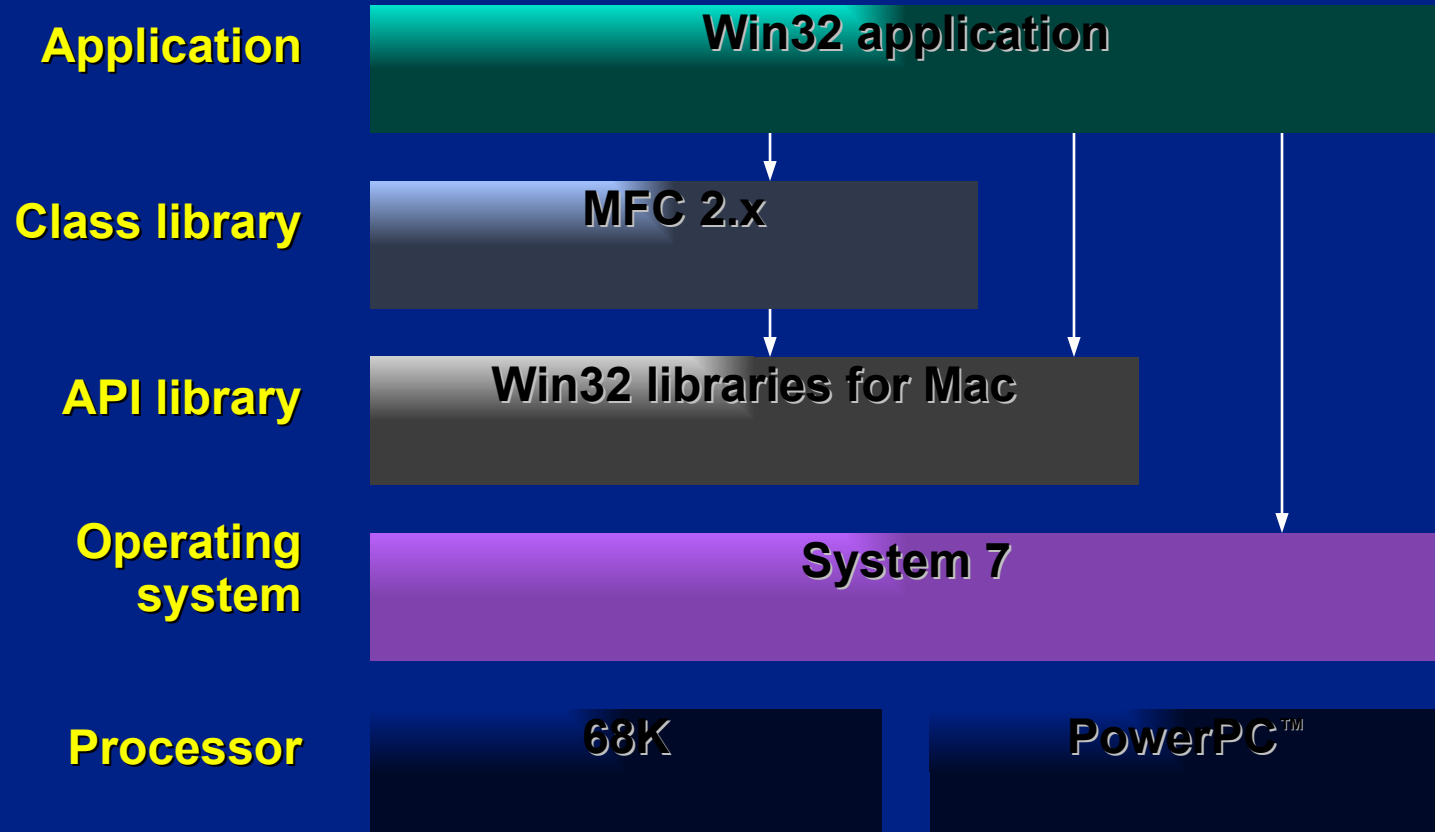
Win32s and MFC



Win32 and MFC



MFC and the Mac



Why Use MFC?

It's the C++ version of the Windows API

**It provides the essential structure
required to build a Windows application.**

MFC apps are fast and relatively small

**MFC apps are portable between
Win16/Win32**

**Using MFC will save you time/effort
building Windows applications.**

Features of MFC

Application Support Classes

Windowing Classes

General Purpose Classes

OLE 2.0 Classes

ODBC Database Classes

MFC Supplies:

- WinMain**

- Message Loop**

- Generic window classes and window procedure**

Class Hierarchy - Application Architecture Classes

For help on a class,
click the class button.

CObject

CCmdTarget

CWinApp

CDocTemplate

CSingleDocTemplate

CMultiDocTemplate

CDocument

CWnd

CView

CScrollView

CFormView

CEditView

CCmdUI

CPrintInfo

Application Support for:

Application initialization

Instance initialization

Instance termination

Dialog data exchange/validation

Commands, command-routing

Context-Sensitive Help

DLL-based apps

Printing/Print Preview

Document/View integration

Class Hierarchy - Visual Object Classes



Windowing Support

Window management

MDI / SDI windows

Graphics, GDI

Menus, controls

Custom dialogs, common dialogs

Scrolling and splitter windows

Toolbars / Status Bars

Edit and form views

Class Hierarchy - General Purpose Classes

CObject

CFile

CStdioFile

CMemFile

CException

CMemoryException

CFileException

CArchiveException

CNotSupportedException

CResourceException

CUserException

COLEException

CByteArray

CWordArray

CDWordArray

CPtrArray

CObArray

CStringArray

CUIntArray

CPtrList

CObList

CStringList

CMapWordToPtr

CMapPtrToWord

CMapPtrToPtr

CMapWordToOb

CMapStringToPtr

CMapStringToOb

CMapStringToString

For help on a class,
click the class button.

CArchive

CMemoryState

CRect

CDumpContext

CString

CPoint

CRuntimeClass

CTime

CSize

CFileStatus

CTimeSpan

General Purpose Support:

Collections: lists, arrays, maps

Strings

Time, date, time span

File access

Object persistence and serialization

Exception handling

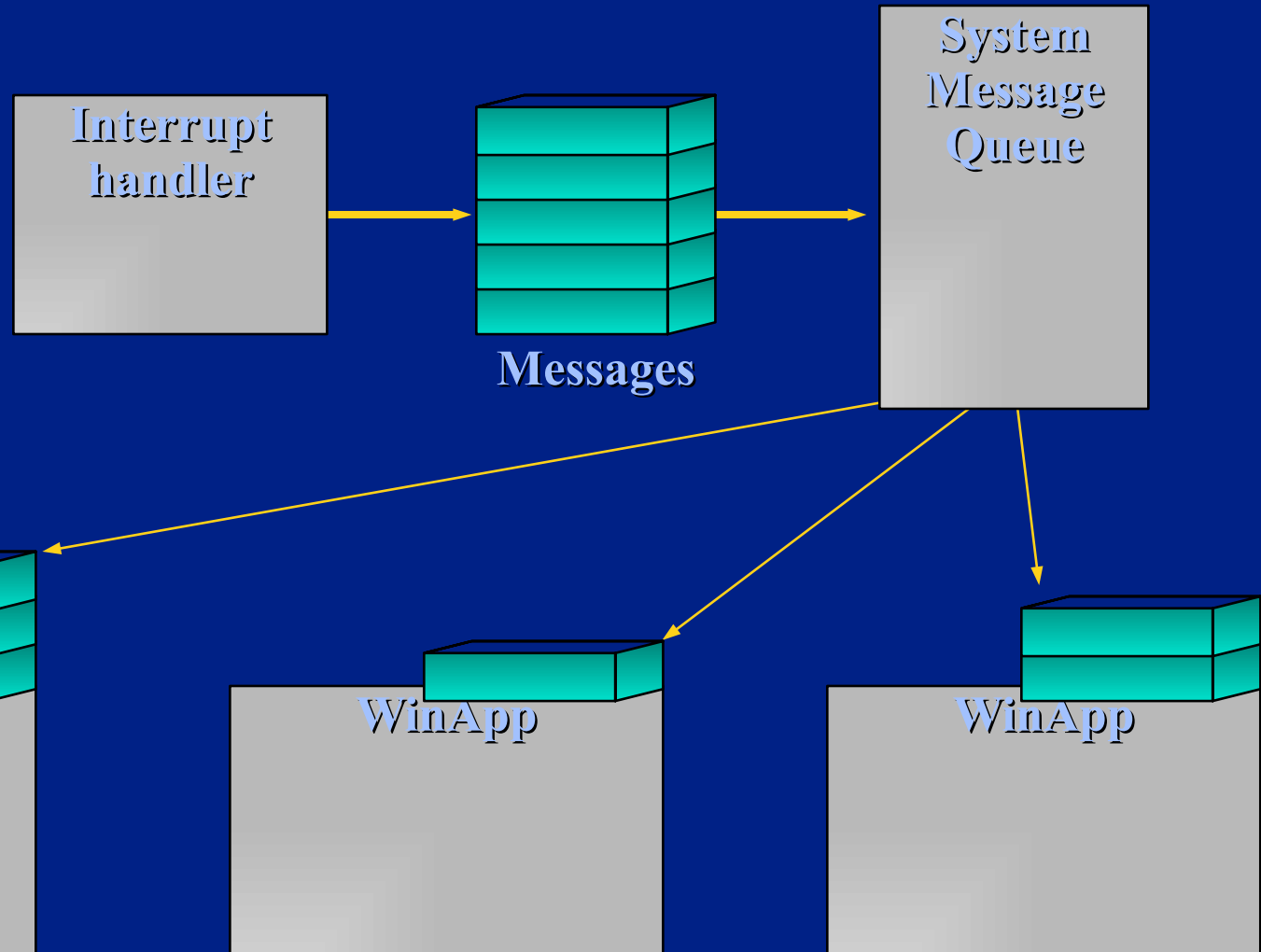
Run-time type info

Windows Messaging

Mouse events



Keyboard
events



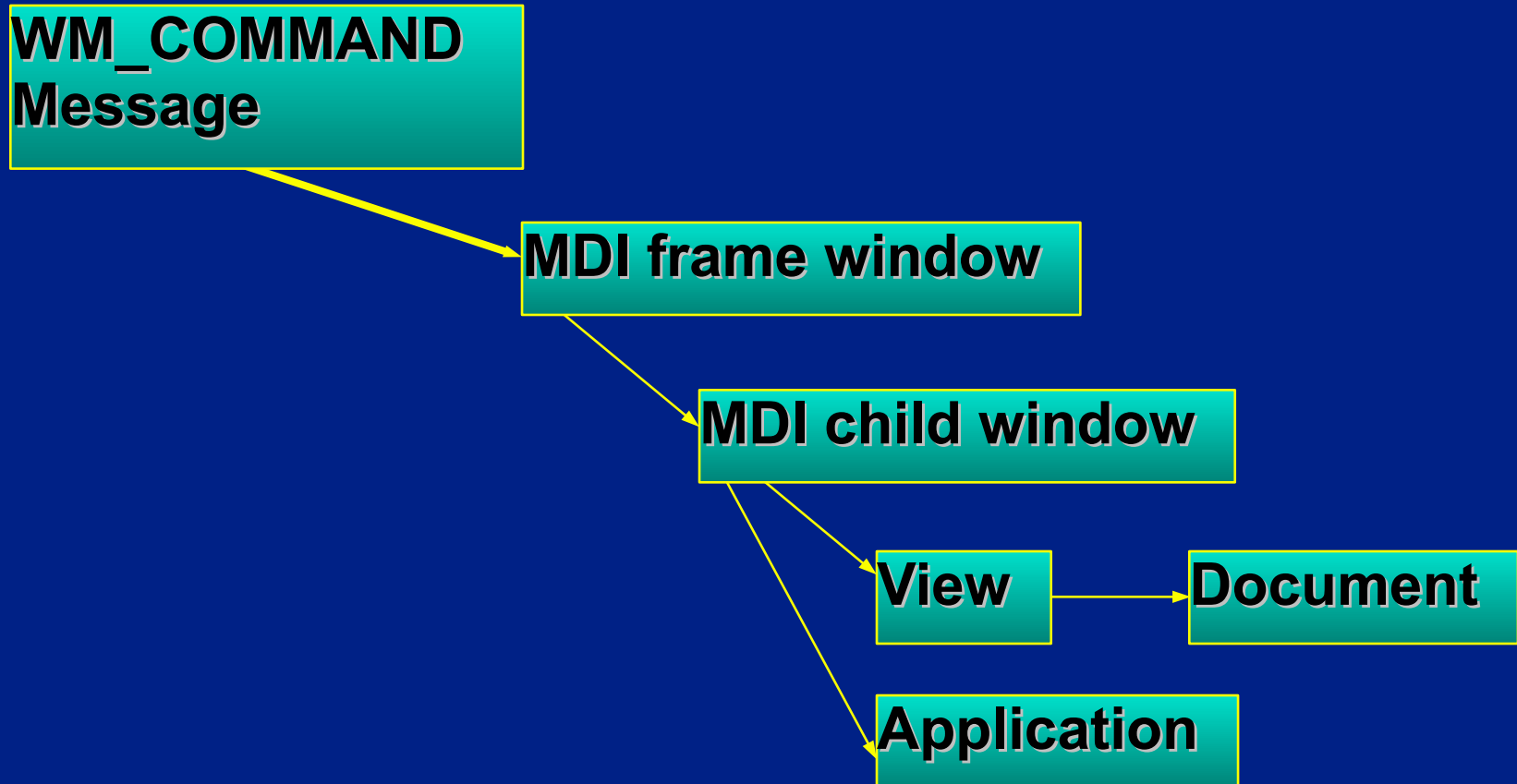
MFC's Message-Mapping

**Routes Windows' messages to the C++
class object managing a window**

**C++ objects map messages to member
functions**

**Thus, member functions are called in
response to messages**

MFC Command Routing



MFC's Document-View Architecture

Documents are something that contain data, typically a file

Views are something that can access, display and edit that data

You can create more than one view of each document

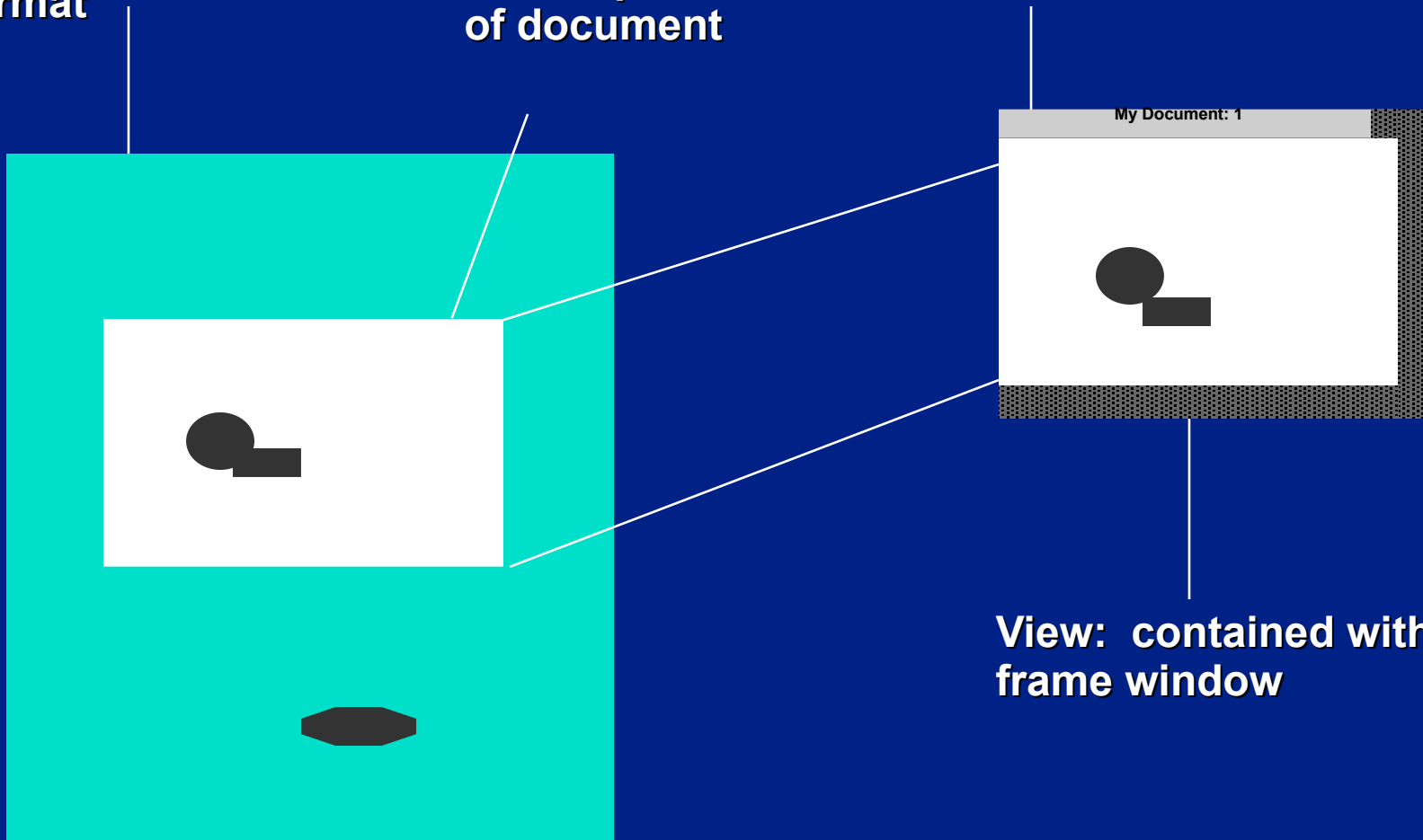
Document-templates let you marry documents, views and windows -- an MFC *menage a trois*

MFC Documents And Views

Document: stores data in efficient format

Visible portion of document

View: renders visible data and responds to user



MFC Documents And Views

CDocument

Manages application-specific data

**Supports disk file, database, OLE
Compound File, non-file-based
storage mechanisms**

**Loads and saves state by overriding
Serialize member function**

MFC handles the standard UI

MFC Documents And Views

CView

CView *is-a* simple child window

Can be anywhere (SDI, MDI, in-place)

Supports multiple (distinct) views on a single document

Replaces low-level OnPaint with abstract OnDraw overridable

Provides change notification and enables optimized drawing

Visual C++ and MFC On the Inside

How MFC Works

A Generic MFC Program: GENERIC.CPP

Modifying GENERIC.CPP

The Generic MFC Program: GENERIC.CPP

Application class object

Instance Initialization

Window class objects

Creating the main window

Displaying the window

Message-maps

What's Provided by MFC:

WinMain

The Message Loop

The Window Procedure

Lot's more!

Pointers to More Information

**VBPJ -- tell Jim and Frank you want more VC+
+/MFC coverage!**

**MFC Journal (mfcj@aol.com -- tell them Richard
sent you)**

Windows Tech Journal (see the *Virtual Column*)

Microsoft Systems Journal (occasional articles)

***Inside Visual C++* from MS Press**

***Visual Programming++* from Addison-Wesley**

Questions...and Answers

An Introduction to Visual C++ and MFC for Visual Basic Developers

by

Richard Hale Shaw

Richard Hale Shaw is a Contributing Editor to PC Magazine, Windows Tech Journal, MFC Journal, Visual Basic Programmer's Journal and Microsoft Systems Journal. He's also the editor of NT Developer Journal and writing *Visual Programming++* for Addison-Wesley. He can be reached at MCIMAIL 399-8368 or CompuServe 72241,155.