MFC

CArchive

CArchive ---- 14
Close ---- 15
Flush( ) ---- 16
MapObject ---- 16
Read ---- 18
ReadClass ---- 19
ReadObject ---- 19
ReadString ---- 20
SerializeClass ---- 21
Write ---- 22
WriteClass ---- 22
WriteObject ---- 23
WriteString ---- 23

CButton

CButton ---- 25
Create ---- 25
GetCursor ---- 26
GetIcon( ) ---- 26
SetButtonStyle ---- 26
SetCursor; ---- 27
SetIcon ---- 27
SetState ---- 28

AddView ---- 48
CDocument ---- 50
DeleteContents ---- 50
GetDocTemplate ---- 51
GetFile ---- 52
GetFirstViewPosition ---- 52
GetNextView ---- 53
GetPathName ---- 53
GetTitle ---- 54
IsModified ---- 54
OnChangedViewList ---- 54
OnCloseDocument ---- 54
OnFileSendMail ---- 55
OnNewDocument ---- 55
OnOpenDocument ---- 56
OnSaveDocument ---- 58
PreCloseFrame ---- 58
ReleaseFile ---- 59
RemoveView ---- 59
SaveModified ---- 59
SetModifiedFlag ---- 60
SetPathName ---- 60
SetTitle ---- 60
UpdateAllViews ---- 60

CEdit

CEdit ---- 63
Clear ---- 63
Create ---- 64
Cut ---- 64
GetFirstVisibleLine ---- 65
GetHandle ---- 65
GetLimitText ---- 66
GetLine ---- 66
GetLineCount ---- 66

GetModify ---- 67
GetPasswordChar ---- 67
GetRect ---- 67
LineFromChar ---- 68
Paste ---- 68
SetHandle ---- 68
SetModify ---- 69
GetPasswordChar ---- 69
SetReadOnly ---- 70
GetRect ---- 70
SetTabStops ---- 71
Undo ---- 72

CFile

CFile ---- 73
Close( ) ---- 75
Duplicate ---- 75
Flush ---- 75
GetFileName ---- 76
GetFilePath ---- 76
GetFileTitle ---- 76
GetLength ---- 77
GetPosition ---- 77
GetStatus ---- 77
LockRange ---- 79
Open ---- 79
Read ---- 81
ReadHuge ---- 81
Remove ---- 82
Rename ---- 82
Seek ---- 83
SeekToBegin ---- 84
SeekToEnd ---- 84
SetFilePath ---- 85
SetLength ---- 85
SetStatus ---- 85

UnlockRange ---- 86
Write ---- 87
WriteHuge ---- 87

CFrameWnd

CFrameWnd ---- 89
Create ---- 90
CreateView ---- 90
GetActiveDocument ---- 91
GetActiveFrame ---- 91
GetActiveView ---- 91
GetControlBar ---- 92
GetMessageString ---- 92
LoadFrame ---- 92
OnCreateClient ---- 93
SetActiveView ---- 93
SetActiveView ---- 94
ShowOwnedWindows ---- 94

CListView

CListView ---- 95
GetListCtrl ---- 95

CObject

AssertValid ---- 96
CObject ---- 97
Dump ---- 98
GetRuntimeClass ---- 98
IsKindOf ---- 99
IsSerializable ---- 100
Serialize ---- 100

CSocket

Attach ---------- 102
CancelBlockingCall ---------- 103
Create---------- 103
CSocket ---------- 104
FromHandle ---------- 104
IsBlocking ---------- 105
OnMessagePending ---------- 105

CSocketFile

CSocketFile---------- 106

CString

Compare ---------- 109
CompareNoCase ---------- 109
CString ---------- 110
Delete----------111
Empty ---------- 112
Find ---------- 112
FindOneOf---------- 113
Format---------- 113
GetAt ---------- 114
GetBuffer ---------- 114
GetBufferSetLength ---------- 115
GetLength---------- 116
Insert---------- 117
IsEmpty ---------- 117
Left---------- 118
LoadString ---------- 118
LockBuffer---------- 119
Mid ---------- 120
ReleaseBuffer ---------- 120

Remove ---------- 121
Replace ---------- 121
ReverseFind ---------- 122
Right ---------- 123
SetAt ---------- 123
TrimLeft ---------- 124
TrimRight ---------- 124
UnlockBuffer ---------- 125

CStringArray

CStringList

CTime

CTime ---------- 128
Format ---------- 130
FormatGmt ---------- 131
GetAsSystemTime ---------- 131
GetCurrentTime ---------- 132
GetDay ---------- 132
GetDayOfWeek ---------- 132
GetGmtTm ---------- 133
GetHour ---------- 134
GetLocalTm ---------- 134
GetMinute ---------- 135
GetMonth ---------- 135
GetSecond ---------- 135
GetTime ---------- 135
GetYear ---------- 136

CView

OnFileOpen ---------- 158
OnFilePrintSetup ---------- 159
OnHelp ---------- 161
OpenDocumentFile ---------- 161
ParseCommandLine ---------- 162
PreTranslateMessage ---------- 162
ProcessMessageFilter ---------- 163
ProcessShellCommand ---------- 163
ProcessWndProcException ---------- 164
RegisterShellFileTypes ---------- 165
Run ---------- 165
RunAutomated ---------- 165
RunEmbedded ---------- 166
SaveAllModified ---------- 166
SelectPrinter ---------- 166
SetDialogBkColor ---------- 167
SetRegistryKey ---------- 167
WriteProfileString ---------- 168

CWindowDC

CWindowDC ---------- 170

CWinThread

CreateThread ---------- 171
CWinThread ---------- 172
ExitInstance ---------- 172
GetMainWnd ---------- 173
GetThreadPriority ---------- 173
InitInstance ---------- 174
PostThreadMessage ---------- 174
PreTranslateMessage ---------- 175
ProcessMessageFilter ---------- 175
ProcessWndProcException ---------- 175

ResumeThread ---------- 176
Run ---------- 176
SetThreadPriority ---------- 177
SuspendThread ---------- 178

CWnd

BeginPaint ---------- 179
BindDefaultProperty ---------- 180
BindProperty ---------- 181
CancelToolTips ---------- 181
CenterWindow ---------- 182
Create ---------- 182
CreateCaret ---------- 183
CreateControl ---------- 183
CWnd ---------- 185
Default ---------- 185
DefWindowProc ---------- 185
DeleteTempMap ---------- 185
DestroyWindow ---------- 186
EndPaint ---------- 186
FindWindow ---------- 187
FlashWindow ---------- 187
GetActiveWindow ---------- 188
GetCurrentMessage ---------- 188
GetDC ---------- 188
GetDesktopWindow ---------- 189
GetFocus ---------- 189
GetIcon ---------- 189
GetNextWindow ---------- 189
GetParent ---------- 190
GetSystemMenu ---------- 190
GetTopWindow ---------- 191
GetWindow ---------- 191
GetWindowDC ---------- 192
GetWindowText ---------- 193
IsChild ---------- 193

KillTimer---------- 193
MoveWindow ---------- 194
OnActivate---------- 194
OnActivateApp ---------- 195
OnCancelMode ---------- 196
OnChar ---------- 196
OnChildNotify ---------- 197
OnClose ---------- 198
OnCommand ---------- 198
OnCreate ---------- 198
OnDestroy ---------- 199
OnEnable ---------- 199
OnKeyDown ---------- 200
OnKeyUp ---------- 201
OnKillFocus ---------- 202
OnLButtonDblClk ---------- 202
OnLButtonDown ---------- 203
OnMButtonDblClk ---------- 204
OnMButtonDown---------- 205
OnMButtonUp ---------- 205
OnMouseActivate;---------- 206
OnMouseMove---------- 206
OnMouseWheel---------- 207
OnMove---------- 208
OnMoving ---------- 208
OnNotify ---------- 209
OnPaint---------- 209
OnRButtonDblClk---------- 210
OnRButtonDown ---------- 211
OnRButtonUp ---------- 211
OnSetCursor---------- 212
OnSetFocus---------- 212
OnShowWindow---------- 213
OnSize---------- 213
OnTimer---------- 214
PostMessage---------- 215
RedrawWindow ---------- 215

ReleaseDC ---------- 216
SendMessage ---------- 217
SendNotifyMessage ---------- 217
SetActiveWindow ---------- 218
SetCapture ---------- 218
SetFocus ---------- 218
SetTimer ---------- 219
SetWindowPos ---------- 219
SetWindowText ---------- 222
ShowWindow ---------- 222
UpdateData ---------- 223
WindowProc ---------- 223

CArchive

CArchive

CArchive

“ ”

/

/ ASCII

CArchive CFile

/

CArchive CFile

CArchive CObject

Serialize DECLARE_SERIAL

IMPLEMENT_SERIAL CObject

>> << CObject

CArchive

pFile CFile CFile

nMode nMode

· CArchive::load CFile

· CArchive::store CFile

· CArchive::bNoFlushOnDelete
Flush Close

nBufSize
4096

lpBuf nBufSize

CArchive

CFile
GetFile
CFile::GetPosition
CArchive::Flush

```
extern char* pFileName;
CFile f;
char buf[512];
if(!f.Open( pFileName,CFile::modeCreate| CFile::modeWrite))
{
  #ifdef_DEBUG
     afxDump<< "Unable to open file"<< "\n"
     exit(1);
  #endif
}
```

Close

Close

CFile::Close CFile

Flush( )

CArchive MapObject
MapObject CArchive
ReadObject WriteObject

```
//MyDoc.h
//Document should have DECLEAR_SERIAL and IMPLEMENT_SERIAL
class CMyDocument: public CDocument
{
  CObList m_listOfSubItems;
  ...
  DECLARE_SERIAL(CMyDocument)
};
//MyDoc.cpp
...
IMPLEMENT_SERIAL(CMyDocument.CObject,1)
...
void CMyDocument::Serialize(CArchive& ar)
{
  if (ar.IsStoring( ))
  {
    //TODO:add storing code here
  }
  else
  {
    //TODO:add loading code here
  }
  ar.MapObject(this);
  //serialize the subitems in the documents;
  //they will be able to serialize their m_pDoc
  //back pointer
  m_listOfSubItems.Serialize(ar);
}
//SubItem.h
class CSubItem:public CObject
{
  public:
   CSubItem(CMYDocument * pdoc) { m_pDoc = pDoc }
```

```
    //back pointer to owning document
    CMYDocument* m_pDoc
    WORD m_I; //other item data
    virtual void Serialize(CArchive& ar);
};
//SubItem.cpp
void CSubItem:Serialize(CArchive& ar)
{
  if (ar.IsStoring( ))
  {
    //will serializing a reference
    //to the “mapped”document pointer
    ar <<M_PDOC;
    ar<<M_I;
  }
  else
  {
    //will load a reference to
    //the “mapped”document pointer
    ar >>m_pDoc;
    ar >>m_i;
  }
}
```

Read

lpBuf

nMax

Serialize Read

```
extern CArchive ar;
char pb[100];
UINT nr=ar.Read(pb.100);
```

ReadClass

CRuntimeClass

pClassRefRequested CRuntimeClass
NULL

pSchema

obTag ReadObject
obTag NULL

WriteClass

pClassRefReguested NULL ReadClass

ReadClass CArchiveException

DECLARE_SERIAL IMPLEMENT_SERIAL ReadClass
CNotSupportedException

pSchema NULL CArchive::GetObjectSchema
*pSchema

SerializeClass ReadClass

ReadObject

CObject CObject::IsKindOf

pClass CRumtimeClass

CArchive >> CObject ReadObject
Serlialize

pClass RUNTIME_CLASS
IMPLEMENT_SERIAL

ReadString

True False

LPTSTR LPTSTR NULL

rString CString CString CArchive

lpsz

nMax lpsz

CArchive nMax
nMax-1
“/0”

CArchive::Read -

SerializeClass

pRuntimeClass

SerializeClass CArchive
CArchive SerializeClass ReadClass
WriteClass SerializeClass

ReadClass SerializeClass
SerializeClass CArchiveException

DECLARE_SERIAL IMPLEMENT_SERIAL, SerializeClass
CNotSupportException

RUMTIME_CLASS pRuntimeClass
IMPLEMENT_SERIAL

```
}
```

Write

lpBuf

nMax

Serialize Write

```
extern CArchive ar;
char pb[100];
ar.Write(pb.100);
```

WriteClass

pClassRef CRuntimeClass

WriteClass WriteClass CRuntimeClass CArchive CArchive::ReadClass

WriteClass WriteClass CArchive::Exception

DECLARE_SERIAL IMPLEMENT_SERIAL WriteClass CNotSuppotedException

SerializeClass WriteClass

WriteObject

pOb

CObject

CObject CArchive << WriteClass Serialize

IMPLEMENT_SERIAL WriteObject ASCII

ASCII

WriteString

lpsz CArchive

‘\0’

WriteString Write

CButton

CButton Windows

push button CButton
Create

CButtonBitmap CButton
CButtonBitmap

CButton CButton
Create Windows CButton

CButton
Create

Windows CDialog

ON_Notification(id, memberFxn)

id ID memberFxn

afx_msg void memberFxn( );

ON_BN_CLICKED

ON_BN_DOUBLECLICKED

CButton

CButton CButton CButton new

delete

CButton

CButton

Create

0

lpszCaption

GetButtonStyle

SetCursor;

hCursor

- BS_TOP
- BS_LEFT
- BS_RIGHT
- BS_CENTER
- BS_BOTTOM
- BS_VCENTER

CBitmapButton SetCursor

SetIcon

hIcon

- BS_TOP
- BS_LEFT
- BS_RIGHT
- BS_CENTER
- BS_BOTTOM
- BS_VCENTER

CbitmapButton

SetState

bHighlight , 0

CClientDC

CClientDC CDC Windows GetDC ReleaseDC CClientDC

CClientDC

pWnd

CClientDC , pWnd CWnd Windows GetDC

Windows GetDC CResourceException Windows

Windows

CCmdTarget

CCmdTarget MFC

CCmdTarget CView CWinApp CDocument CWnd CFrameWnd

CCmdTarget

OnCmdMsg “VisualC++ ” “ ” “ ” “ ”

CmdTarget

OLE IDispatch

VB IDispatch “Win32 SDK OLE ” “ IDPatch ” “ API ”

BeginWaitCursor

CDocument

BeginWaitCursor OnSetCursor EndWaitCursor

```
//The following example illustrates the most common case
//of displaying the hourglass cursor during some lengthy
//processing of a command handler implemented in some
//CCmdTaget-derived class,such as a document or view.
```

```
void CMyView::OnSomeCommand( )
{
  BeginWaitCursor( ); //display the hourglass cursor
  //do some lengthy processing
  EndWaitCursor( ); //remove the hourglass cursor
}
//The next example illustrates RestoreWaitCursor
void CMyView::OnSomeCommand( )
{
  BeginWaitCursor( ); //display the hourglass cursor
  //do some lengthy processing
  // The dialog box will normally change the cursor to
  // the standard arrow cursor,and leave the cursor in
  // as the standard arrow cursor when the dialog box is
  //closed.
  CMyDialog dlg;
  dlg.DoModal( );
  // It is necessary to call RestoreWaitCursor here in order
  // to change the cursor back to the hourglass cursor.
  RestoreWaitCursor( );
  // do some more lengthy processing
  EndWaitCursor( ); //remove the hourglass cursor
}
```

```
    //some more processing
}
```

```
}
// The next example illustrates RestoreWaitCursor
void CMyView::OnSomeCommand( )
{
  BeginWaitCursor( ); // display the hourglass cursor
  // do some lengthy processing
  // The dialog box will normally change the cursor to
  // the standard arrow cursor,and leave the cursor in
  // as the standard arrow cursor when the dialog box is
  // closed.
  CMyDialog dlg;
  dlg.DoModal( );
  // It is necessary to call RestoreWaitCursor here in order
  // to change the cursor back to the hourglass cursor.
  RestoreWaitCursor( );
  // do some more lengthy processing
  EndWaitCursor( ); // remove the hourglass cursor
}
// In the above example,the dialog was clearly invoked between
// the pair of calls to BeginWaitCursor and EndWaitCursor.
// Sometimes it may not be clear whether the dialog is invoked
// in between a pair of calls to BeginWaitCursor and EndWaitCursor.
// It is permissable to call RestoreWaitCursor,even if
// BeginWaitCursor was not previously called.This case is
// illustrated below,where CMyView::AnotherFunction does not
// need to know whether it was called in the context of an
// hourglass cursor.
void CMyView::AnotherFunction( )
{
  // some processing...
  MyDialog dlg;
  dlg.DoModal( );
  RestoreWaitCursor( );
  // some more processing ...
}
// If the dialog is invoked from a member function of
// some non-CCmdTarget,the you call CWinApp::DoWaitCursor
// with a 0 parameter value to restore the hourglass cursor.
```

```
void CMyObject::AnotherFunction( )
{
  CMyDialog dlg;
  dlg.DoModal( );
  AfxGetApp( )->DoWaitCursor(0); // same as CCmdTarget::RestorWaitCursor
}
```

FromIDispatch

lpDistpatch CCmdTarget IDistpatch MFC IDistpatch NULL

lpDispatch IDistpatch

IDistpatch IDistpatch CCmdTarget

GetIDispatch

GetIDispatch

IDistpatch

bAddRef

IDistpatch IDistpatch IDistpatch

EnableAutomation IDistpatch MFC IDistpatch IDistpatch

IUnknow::AddRef

IsResultExpected

0

OLE MFC
MFC IsResultExpected

0

OnCmdMsg

0

nID ID

nCode

pExtra nCode

pHandlerInfo OnCmdMsg pHandlerInfo pTarget pmf
NULL

OnCmdMsg CCmdTarget::OnCmdMsg

“Visual C++ ” “ ” “ ”

MFC

“ 21”

```
//This example illustrates extending the framework's standard command
//route from the view to objects managed by the view. This example
//is from an object-oriented drawing application, similar to the
//DRAWCLI sample application, which draws and edits "shapes".
BOOL CMyView::OnCmdMsg(UINT nID, int nCode, void* pExtra,
                       AFX_CMDHANDLERINFO* pHandlerInfo )
{
  //Extend the framework's command route from the view to
  //the application-specific CMyShape that is currently selected.
  //in the view. m_pActiveShape is NULL if no shape object
  //is currently selected in the view.
  if( (m_pActiveShape!=NULL)&&
      m_pActiveShape->OnCmdMsg(nID, nCode, pExtra, pHandlerInfo
    ) )
  return TRUE;
  //If the object(s) in the extended command route don't handle
  //the command, then let the base class OnCmdMsg handle it.
  return CView::OnCmdMsg(nID, nCode, pExtra, pHandlerInfo );
}
//The command handler for ID_SHAPE_COLOR (menu command used to
change
//the color of the currently selected shape ) was added to
//the message map of CMyShape (note, not CMyView ) using ClassWizzard.
//The menu item will be automatically enabled or disabled, depending
//on whether a CMyShape is currently selected in the view, that is
//depending on whether CMyView::m_pActiveView is NULL. It is not
//necessary to implement an ON_UPDATE_COMMAND_UI handler to enable
//or disable the menu item.
BEGIN_MESSAGE_MAP( CMyShape, CCmdTarget )
//{{AFX_MSG_MAP( CMyShape )
   ON_COMMAND( ID_SHAPE_COLOR, OnShapeColor )
//}}AFX_MSG_MAP( )
END_MESSAGE_MAP( )
```

OnFinalRelease

OLE OLE

RestoreWaitCursor

```
// The following example illustrates the most common case
// of displaying the hourglass cursor during some lengthy
// processing of a command handler implemented in some
// CCmdTarget-derived class,such as a document or view.
void CMyView::OnSomeCommand( )
{
  BeginWaitCursor( ); // display the hourglass cursor
  // do some lengthy processing
  EndWaitCursor( ); // remove the hourglass cursor
}
// The next example illustrates RestoreWaitCursor
void CMyView::OnSomeCommand( )
{
  BeginWaitCursor( ); // display the hourglass cursor
  // do some lengthy processing
  // The dialog box will normally change the cursor to
  // the standard arrow cursor,and leave the cursor in
  // as the standard arrow cursor when the dialog box is
  // closed.
  CMyDialog dlg;
  dlg.DoModal( );
  // It is necessary to call RestoreWaitCursor here in order
```

```
  // to change the cursor back to the hourglass cursor.
  RestoreWaitCursor( );
  // do some more lengthy processing
  EndWaitCursor( ); // remove the hourglass cursor
}
// In the above example,the dialog was clearly invoked between
// the pair of calls to BeginWaitCursor and EndWaitCursor.
// Sometimes it may not be clear whether the dialog is invoked
// in between a pair of calls to BeginWaitCursor and EndWaitCursor.
// It is permissable to call RestoreWaitCursor,even if
// BeginWaitCursor was not previously called.This case is
// illustrated below,where CMyView::AnotherFunction does not
// need to know whether it was called in the context of an
// hourglass cursor.
void CMyView::AnotherFunction( )
{
  // some processing...
  CMyDialog dlg;
  dlg.DoModal( );
  RestoreWaitCursor( );
  // some more processing...
}
// If the dialog is invoked from a member function of
// some non-CCmdTarget,the you call CWinApp::DoWaitCursor
// with a 0 parameter value to restore the hourglass cursor.
void CMyObject::AnotherFunction( )
{
  CMyDialog dlg;
  dlg.DoModal( );
  AfxGetApp( )->DoWaitCursor(0); // same as CCmdTarget::RestorWaitCursor
}
```

CDialog

CDialog

CDialog CDialog
ClassWizard CDialog

Windows
ClassWizard
ClassWizard

ClassWizard

ClassWizard
ClassWizard

ClassWizard

DoModal
Create

DLGTEMPLATE “Win32 SDK”
CDialog CreateIndirect
InitModalIndirect DoModal

ClassWizard CWnd::DoDataExchange
CWnd DoDataExchange

CWnd::UpdateData DoDataExchange OK
UpdateData Cancel

OnInitDialog OnInitDialog

OnInitDialog

CWnd::UpdateData

ClassWizard “Visual C++ ” “ ClassWizard”

CWinApp::SetDialogBkColor

OK Cancel EndDialog

OnCancel DestroyWindows CDialog::OnCancel EndDialog PostNcDestory PostNcDestory

CDialog

lpszTemplateName

nIDTemplate ID

pParentWnd NULL

IDD_ IDD_DIALOG1 ID

InitModalIndirect

DoModal

CDialog

Create

CreateIndirect

Create

0

lpszTemplateName

pParentWnd NULL

nIDTemplate ID

Create

Create

ID IDD_DIALOG1

pParentWnd NULL

Create

WS_VISIBLE

ShowWindow “Win32 SDK”

DIGTEMPLATE “Microsoft Visual C++ 6.0 MFC ” “Windows ” CWnd::DestroyWindow Create

DoModal

CDialog::EndDialog nResult

-1 IDABORT

OK Cancel OnOK
OnCancel OnOK
IDOK OnCancel IDCANCEL

PreTransMessage

EndDialog

nResult DoModal

nResult
EndDialog

EndDialog OnInitDialog

EndDialog

NextDlgCtrl

OnCancel

Cancel ESC

Cancel	EndDialog

OnSetFont

pFont

PrevDlgCtrl

SetDefID

nID ID

SetHelpID

nIDR ID

ID

CDocument

CDocument File Open File Save

CDocument CDocument

CDocTemplate

CView

FileSave

CDocument UpdateAllViews

· CDocument

·

·

· CObject::Serialize

CDocument MAPI CDocument “Visual C++ ” “ ( )” “ / ” “ / ”

AddView

pView

File Open FileNew Windows

CDocTemplate

```
// The following example toggle two views in an SDI(single document
// interface) frame window.A design decision must be made as to
// whether to leave the inactive view connected to the document,
// such that the inactive view continues to receive OnUpdate
// keep the inactive view continuously in sync with the document,even
//though it is inactive However,doing so incurs a performance cost,
// as well as the programming cost of implementing OnUpdate hints.It may be less expensive,in terms of performance and/or programming,
// to re-sync the inactive view with the document only with it is
// reactivated. This example illustrates this latter approach,by
// reconnecting the newly active view and disconnecting the newly
// inactive view,via calls to CDocument::AddView and RemoveView.
BOOL CMainFrame::OnViewChange(UNIT nCmdID)
{
  CView* pViewAdd;CView* pViewRemove;
  CDocument* pDoc = GetActiveDocument();
  UNIT nCmdID;nCmdID = LOWORD(GetCurrentMessage()->wParam);
  if ((nCmdID = = ID_VIEW_VIEW1)&&(m_currentView = =1)) return;
  if ((nCmdID = = ID_VIEW_VIEW2)&&(m_currentView = =2))return;
  if (nCmdID = = ID_VIEW_VIEW2)
  {
    if (m_pView2 = = NULL)
    {
```

```
    m_pView1 = GetActiveView();
    m_pView2 = new CMyView2;
    //Note that if onSize has beenoverridden in CMy View2
    //and GetDoument()is used in this overrideit can
    //cause assertions and,if theassertions are ignored,
    //cause access violation.
    m_pView2->Create(NULL, NULL, AFX_WS_DEFAULT_VIEW, rectDefault,
                this, AFX_IDW_PANE_FIRST+1, NULL
               );
  }

  pViewAdd = m_pView2;pViewRemove = m_pView1;
  m_currentView = 2
}
else
{
   pViewAdd = m_pView1;
   pViewRemove = m_pView2;
   m_currentView = 1;
}
// Set the child i.d. of the active view to AFX_IDW_PANE_FIRST,
// so that CFrameWnd::RecalcLayout will allocate to this
// "first pane" that portion of the frame window's client area
// not allocated to control bars.Set the child i.d. of the
// other view to anything other than AFX_IDW_PANE_FIRST;this
// examples switches the child id's of the two views.

int nSwitchChildID = pViewAdd->GetDlgCtrlID();
pViewAdd->SetDlgCtrlID(AFX_LDW_PANE_FIRST);
pViewRemove->SetDlgCtrlID(nSwitchChildID);

// Show the newly active view and hide the inactive view.

pViewAdd->ShowWindow(SW_SHOW);
pViewRemove->ShowWindow(SW_HIDE);

// Connect the newly active view to the document,and
// disconnect the inactive view.
```

```
    pDoc->AddView(pViewAdd);
    pDoc->RemoveView(pViewRemove);
    SetActiveView(pViewAdd);
    RecalcLayout();
  }
```

AddView

pView

File Open FileNew Windows

CDocTemplate

```
  // The following example toggle two views in an SDI(single document
  // interface) frame window.A design decision must be made as to
  // whether to leave the inactive view connected to the document,
  // such that the inactive view continues to receive OnUpdate
  // keep the inactive view continuously in sync with the document,even
  //though it is inactive However,doing so incurs a performance cost,
  // as well as the programming cost of implementing OnUpdate hints.It may be less expensive,in terms of performance and/or programming,
```

```
UNIT nCmdID;nCmdID = LOWORD(GetCurrentMessage()->wParam);
if ((nCmdID = = ID_VIEW_VIEW1)&&(m_currentView = =1)) return;
if ((nCmdID = = ID_VIEW_VIEW2)&&(m_currentView = =2))return;
if (nCmdID = = ID_VIEW_VIEW2)
{
  if (m_pView2 = = NULL)
  {
    m_pView1 = GetActiveView();
    m_pView2 = new CMyView2;
    //Note that if onSize has beenoverridden in CMy View2
    //and GetDoument()is used in this overrideit can
    //cause assertions and,if theassertions are ignored,
    //cause access violation.
    m_pView2->Create(NULL, NULL, AFX_WS_DEFAULT_VIEW, rectDefault,
                this, AFX_IDW_PANE_FIRST+1, NULL
               );
  }

  pViewAdd = m_pView2;pViewRemove = m_pView1;
  m_currentView = 2
}
else
{
   pViewAdd = m_pView1;
   pViewRemove = m_pView2;
   m_currentView = 1;
}
// Set the child i.d. of the active view to AFX_IDW_PANE_FIRST,
// so that CFrameWnd::RecalcLayout will allocate to this
// "first pane" that portion of the frame window's client area
// not allocated to control bars.Set the child i.d. of the
// other view to anything other than AFX_IDW_PANE_FIRST;this
// examples switches the child id's of the two views.

int nSwitchChildID = pViewAdd->GetDlgCtrlID();
pViewAdd->SetDlgCtrlID(AFX_LDW_PANE_FIRST);
pViewRemove->SetDlgCtrlID(nSwitchChildID);
```

```
    // Show the newly active view and hide the inactive view.

    pViewAdd->ShowWindow(SW_SHOW);
    pViewRemove->ShowWindow(SW_HIDE);

    // Connect the newly active view to the document,and
    // disconnect the inactive view.

    pDoc->AddView(pViewAdd);
    pDoc->RemoveView(pViewRemove);
    SetActiveView(pViewAdd);
    RecalcLayout();
}
```

CDocument

CDocument OnNewDocument
SDI

DeleteContents

CDocument
NULL SDI

“Edit Clear All”

```
// This example is the handler for an Edit Clear All Command.
void CMyDoc::OnEditClearAll()
{
  DeleteContents();
  UpdateAllViews(NULL);
```

```
}

void CMyDoc::DeleteContents()
{
  // Reinitialize document data here.
}
```

GetDocTemplate

NULL

```
// This example accesses the doc template object to construct
// a default document name such as SHEET.XLS,where "sheet"
// is the base document name and *.xls* is the file extension
// for the document type.
CString strDefaultDocName, strBaseName, strExt;
CDocTemplate* pDocTemplate = GetDocTemplate();
if (!pDocTemplate->GetDocString(strBaseName,CDocTemplate::docName) ||
    !pDocTemplate->GetDocString(strExt, CDocTemplate::filterExt)
   )
{
  AfxThrowUserException();
  // These doc template strings will
  // be available if you created the application using AppWizard
  // and specified the file extension as an option for
  // the document class produced by AppWizard.
}
strDefaultDocName = strBaseName + strExt;
```

GetFile

CFile

lpszFileName

pError

nOpenFlags OR

CFile CFile::CFile ,

modeCreate modeNoInherit

CFile

GetFirstViewPosition

POSITION GetNextView

```
// To get the first view in the list of views:
POSITION pos = GetFirstViewPositon ();
CView* pFirstView = GetNextView( pos );
// This example uses CDocument::GetFirstViewPosition
// and GetNextView to repaint each view.
void CMyDoc::OnRepaintAllViews()
{
  POSITION pos = GetFirstViewPosition();
  while (pos != NULL)
  {
    CView* pView = GetNextView(pos);
    pView->UpdateWindow();
  }
```

```
}
// An easier way to accomplish the same result is to call
// UpdateAllViews(NULL);
```

GetNextView

rPosition

rPosition POSITION GetNextView
GetFirstViewPosition NULL

rPosition rPosition
POSITION rPosition
NULL

```
// This example uses CDocument::GetFirstViewPosition
// and GetNextView to repaint each view.
void CMyDoc::OnRepaintAllViews()
{
  POSITION pos = GetFirstViewPosition();
  while (pos != NULL)
  {
    CView* pView = GetNextView(pos);
    pView->UpdateWindow();
  }
}
// An easier way to accomplish the same result is to call
// UpdateAllViews(NULL);
```

GetPathName

GetTitle

IsModified

0

OnChangedViewList

/

OnCloseDocument

File Close

DeleteContents

OnFileSendMail

OnFileSendMail OnSave-Document

MAPI32. DLL OnFileSendMail

COleDocument OnFileSendMail

OnFileSendMail

“Visual C++ ” “MFC MAPI

”

OnNewDocument

0

File New

SDI File New

MDI File New

FileNew SDI

```
// The following examples illustrate alternative methods of
// initializing a document object.
// Method 1: In an MDI application,the simplest place to do
// initialization is in the document constructor.The framework
// always creates a new document object for File New or File Open.
CMyDoc::CMyDoc()
{
  // Do initialization of MDI document here.
  // ...
```

```
}

// Method 2: In an SDI or MDI application,do all initialization
// in an override of OnNewDocument,if you are certain that
// the initialization is effectively saved upon File Save
// and fully restored upon File Open,via serialization.
BOOL CMyDoc::OnNewDocument()
{
  if (!CDocument::OnNewDocument()) return FALSE;
  // Do initialization of new document here.
  return TRUE;
}

// Method 3: If the initialization of your document is not
// effectively saved and restored by serialization(during File Save
// and File Open),the implement the initialization in single
// function(named InitMyDocument in this example).Call the
// shared initialization function from override of both
// OnNewDocument and OnOpenDocument.
BOOL CMyDoc::OnNewDocument()
{
  if (!CDocument::OnNewDocument()) return FALSE;
  InitMyDocument(); //call your shared initialization function
  // If your new document object requires additional initialization
  // not necessary when the document is deserialized via File Open,
  // then perform that additional initialization here.
  return TRUE;
}
```

OnOpenDocument

0

lpszPathName

File Open

DeleteContents CObject::Serialize

SDI File Open CDocument

MDI File Open

CDocument

File Open SDI

```
// The following examples illustrate alternative methods of
// initializing a document object.
// Method 1: In an MDI application,the simplest place to do
// initialization is in the document constructor.The framework
// always creates a new document object for File New or File Open.
CMyDoc::CMyDoc()
{
  // Do initialization of MDI document here.
  // ...
}

// Method 2: In an SDI or MDI application,do all initialization
// in an override of OnNewDocument,if you are certain that
// the initialization is effectively saved upon File Save
// and fully restored upon File Open,via serialization.
BOOL CMyDoc::OnNewDocument()
{
  if (!CDocument::OnNewDocument()) return FALSE;
  // Do initialization of new document here.
  return TRUE;
}

// Method 3: If the initialization of your document is not
// effectively saved and restored by serialization(during File Save
// and File Open),the implement the initialization in single
// function(named InitMyDocument in this example).Call the
// shared initialization function from overrides of both
```

```
// OnNewDocument and OnOpenDocument.
BOOL CMyDoc::OnNewDocument()
{
  if (!CDocument::OnNewDocument()) return FALSE;
  InitMyDocument(); //call your shared initialization function
  // If your new document object requires additional initialization
  // not necessary when the document is deserialized via File Open,
  // then perform that additional initialization here.
  return TRUE;
}
```

OnSaveDocument

0

lpszPathName

File Save File Save As

CObject::Serialize

PreCloseFrame

pFrame CDocument CFrameWnd

PreCloseFrame CDocument CDocument COleDocument CRichEditDoc

ReleaseFile

pFile CFile

bAbort CFile::Close CFile::Abort CFile::Close
FALSE CFile::Abort FALSE

bAbort TRUE ReleaseFile
CFile::Abort CFile::Abort bAbort FALSE ReleaseFile
CFile::Close

RemoveView

pView

NULL

CDocTemplate

SaveModified

0

SetModifiedFlag

bModified

bModified TRUE FALSE

SetPathName

lpszPathName

bAddToMRU MRU TRUE
FALSE

MRU
bAddToMRU

SetTitle

lpszTitle

UpdateAllViews

pSender NULL

lHint

pHint

SetModifiedFlag pSender

CView::OnUpdate lHint pHint
lHint

CObject pHint
CView CView::OnUpdate

CEdit

CEdit Windows

CEdit

CEdit Create Windows CEdit

CEdit

Create

CEdit CWnd CEdit CWnd

SetWindowText GetWindowText

CEdit GetLine GetSel GetSel

ReplaceSel

CDialog

ON_Notification(id,memberFxn)

id ID memberFxn

afx_msg void memberFxn();

· ON_EN_CHANGE

EN_UPDATE Windows

· ON_EN_ERRSPACE

- ON_EN_HSCROLL

- ON_EN_KILLFOCUS

- ON_EN_MAXTEXT
  ON_EN_HSCROLL
  ON_EN_VSCROLL

- ON_EN_SETFOCUS

- ON_EN_UPDATE

- ON_EN_VSCROLL

CEdit CEdit

CEdit CEdit

CEdit CEdit

new CEdit delete

CEdit CEdit

CEdit

CEdit Create Windows

Clear

Clear Undo

Cut

Win 32 WM_CLEAR

Create

0

dwStyle

rect RECT CRect

pParentWnd CDialog NULL

nID ID

CEdit CEdit Create
Windows CEdit Create Windows
WM_NCCREATE WM_NCCALCSIZE WM_CREATE WM_GETMINMAXINFO

CWnd OnNcCalcSize OnCreate OnNcCreate
OnGetMinMaxInfo CEdit
OnCreate

- WS_CHILD
- WS_VISIBLE
- WS_DISABLED
- WS_GROUP
- WS_TABSTOP

Cut

CF_TEXT

Cut Undo

Clear

Win 32 WM_CUT

GetFirstVisibleLine

0 0

Win32 EM_GETFIRSTVISIBLELINE

GetHandle

0

Windows

GetHandle

DS_LOCALEDIT

GetHandle Windows 95 Windows 95 NULL GetHandle Windows NT 3.51

Win32 EM_GETHANDLE

GetLimitText

CEdit

CEdit

Windows 95 Windows NT 4.0

Win32 EM_GETLIMITTEXT

GetLine

nIndex 0

nIndex 0

lpszBuffer

nMaxLength GetLine Windows
lpszBuffer

lpszBuffer

Win32 EM_GETLINE

GetLineCount

1

GetModify

0

Windows

SetModify

Win32 EM_GETMODIFY

GetPasswordChar

NULL

ES_PASSWORD *

Win32 EM_GETPASSWORDCHAR

GetRect

lpRect RECT

SetRect SetRectNP

Win32 EM_GETRECT

LineFromChar

nIndex 0 nIndex -1

nIndex 0 -1 -1

Win32 EM_LINEFROMCHAR

Paste

CEdit

CF_TEXT

Win32 WM_PASTE

SetHandle

hBuffer LocalAlloc Windows
LMEM_MOVEABLE
0

GetHandle
LocalFree Windows
CanUndo 0 GetModify
0
DS_LOCALEDIT

Windows 95 Windows 95 GetHandle NULL
Windows NT 3.51
Win32 EM_SETHANDLE LocalAlloc LocalFree

SetModify

bModified TRUE FALSE

GetModify
Win32 EM_SETMODIFY

GetPasswordChar

ch 0

SetPasswordChar CEdit ch

ES_PASSWORD

* SetPasswordChar ch=0

Win32 ES_SETPASSWORDCHAR

SetReadOnly

0

bReadOnly TRUE

FALSE

CWnd::GetStyle

ES_READONLY Win32

EM_SETREADONLY

GetRect

lpRect RECT CRect

SetRect

SetRect

GetRect

SetRect

Win32 EM_SETRECT

SetTabStops

0

cxEachStop cxEachStop

nTabStops rgTabStops 1

rgTabStops

WindowsGetDialogBaseUnits

32 32

cxEachStop

SetTabStops

CWnd::InvalidateRect

Win32 EM_SETTABSTOPS

Undo

0

Win32 EM_UNDO

CFile

CFile MFC /

CFile CArchive MFC

CFile

CFile I/O ofstream Microsoft

CFile

CFile “Visual C++ ” “MFC ” “Microsoft Visual C++ 6.0 ” “ ”

CFile

hFile

lpszFileName

nOpenFlags OR(|)

modeCreate modeNoInherit

· CFile::modeCreate

0

· CFile::modeNoTruncate modeCreate

0

CStdioFile.

- CFile::modeRead
- CFile::modeReadWrite
- CFile::modeWrite
- CFile::modeNoInherit
- CFile::ShareDenyNone
  Create
- CFile::ShareDenyRead
  Create
- CFile::ShareDenyWrite
  Create
- CFile::ShareExclusive
- CFile::ShareCompat 32 MFC CFile::Open CFile::ShareExclusive
- CFile::typeText
- CFile::typeBinary

m_hFile CFile::hFileNull
TRY/CATCH Open
“Visual C++ ” “ ”

CFile hFile
CFile

CFile
Open

```
// example for CFile::CFile
```

```
char* pFileName = "test.dat";
TRY
{
  CFile f(pFileName, CFile::modeCreate | CFile::modeWrite);
}

CATCH(CFileException,e)
{
  #ifdef _DEBUG
     afxDump<<"File could not be opened"<<e->m_cause<<"\n";
  #endif
}

END_CATCH
```

Flush CArchive
CArchive::Flush

GetFileName

GetFileName
c:\windows\write\myfile.wri myfile.wri

GetFilePath myfile
GetTitle

GetFilePath

GetFilePath
c:\windows\write\myfile.wri c:\windows\write\myfile.wri

myfile.wri , GetFileName myfile
GetFileTitle

GetFileTitle

GetFileTitle
c:\windows\write\myfile.wri myfile

Windows95 Win32 GetFileTitle

GetFilePath myfile.wri
GetFileName

GetLength

GetPosition

32

Seek

```
// example for CFile::GetPosition
extern CFile cfile;
DWORD dwPosition = cfile.GetPosition( );
```

GetStatus

TRUE FALSE

rStatus CFileStatus CFileStatus

- CTime m_ctime
- CTime m_mtime
- CTime m_atime
- LONG m_size DIR
- BYTE m_attribute
- Char m_szFullName[_MAX_PATH] Windows

lpszFileName Windows

GetStatus CFile m_szFullName

m_szFullName

m_attribute MFC enum

enum Attribute

{

normal = 0x00,

readOnly = 0x01,

hidden = 0x02,

system = 0x04,

volume = 0x08,

directory = 0x10,

archive = 0x20

};

```
// example for CFile::GetStatus
CFileStatus status;
Extern CFile cfile;
If(cfile.GetStatus(status)) // virtual member function
{
  #ifdef _DEBUG
     afxDump<<"File size ="<<status.m_size<<"\n";
  #endif
}
char* pFileName ="test.dat";
if(CFile::GetStatus(pFileName,status)) // status function
```

```
{
  #ifdef _DEBUG
    afxDump<<"Full File name ="<<status.m_szFullName <<"\n";
  #endif
}
```

LockRange

dwPos

dwCount

UnlockRange

LockRange

CMemFile

```
// example for CFile::LockRange
extern DWORD dwPos;
extern DWORD dwCount;
extern CFile cfile;
cfile.LockRange(dwPos,dwCount);
```

Open

0 pError 0

lpszFileName UNC

nOpenFlags UINT
OR | modeCreate
modeNoInherit CFile

pError

Open CFile

CFile Open FALSE Open
CFileException pError NULL
pError Open FALSE CFileException
CFileException Open Open

Open

pError CFileException

NULL No TRUE n/a

ptr CFileException No TRUE

NULL Yes FALSE n/a

ptr CFileException Yes FALSE

```
// example for CFile::Open
CFile f;
CFileException e;
char* pFileName = "test.dat";
if(!f.Open(pFileName, CFile::modeCreate | CFile::modeWrite,&e))
{
  #ifdef _DEBUG
    afxDump<<"File could not be opened"<<e.m_cause<<"\n";
  #endif
}
```

Read

CFile nCount

lpBuf

nCount

CFile

```
// example for CFile::Read
extern CFile cfile;
char pbuf[100];
UINT nBytesRead = cfile.Read(pbuf,100);
```

ReadHuge

CFile

dwCount

lpBuf

dwCount

CFile

Read 64K-1 ReadHuge

CFile

ReadHuge Win32 ReadHuge Read

Remove

lpszFileName

DEL

```
// example for CFile::Remove
char* pFileName ="test.dat";
TRY
{
  CFile::Remove(pFileName);
}

CATCH(CFileException,e)
{
  #ifdef _DEBUG
     afxDump <<"File"<<pFileName<<"cannot be removed\n";
  #endif
}
END_CATCH
```

Rename

lpszOldName

lpszNewName

REN

```
// example for CFile::Rename
extern char* pOldName;
extern char* pNewName;
TRY
{
  CFile::Rename(pOldName,pNewName);
}

CATCH(CFileException,e)
{
  #ifdef _DEBUG
    afxDump << "File" << pOldName << "not found,cause =" << e->m_cause
<<"\n";
  #endif
}
END_CATCH
```

Seek

Seek

CFileException

lOff

nFrom

- CFile::begin lOff
- CFile::current lOff
- CFile::end lOff
  lOff

Seek

0

```
// example for CFile::Seek
extern CFile cfile;
LONG lOffset = 1000;
LONG lActual = cfile.Seek(lOffset, CFile::begin);
```

SeekToBegin

Seek(0L, CFile::begin)

```
// example for CFile::SeekToBegin
extern CFile cfile;
cfile.SeekToBegin( );
```

SeekToEnd

CFile::Seek(0L, CFile::End);

SetFilePath

lPszNewName

CFile

SetFilePath CFile

SetLength

dwNewLen

CMemFile , CMemeryException

```
// example for CFile::SetLength
extern CFile cfile;
DWORD dwNewLength = 10000;
cfile.SetLength(dwNewLength);
```

SetStatus

lpszFileName

status CFileStatus
0 GetStatus
CFileStatus

Status m_mtime

SetStatus

```
cfile.UnlockRange(dwPos,dwCount);
```

Write

lpBuf

nCount

CFile

Write

```
// example for CFile::Write
extern CFile cfile;
char pbuf[100];
cfile.Write(pbuf,100);
```

WriteHuge

lpBuf

dwCount

CFile WriteHuge

Write 64K-1 WriteHuge
CFile

WriteHuge Win32 WriteHuge Write

CFrameWnd

CFrameWnd Windows

CFrameWnd

· Create

· LoadFrame

·

Create LoadFrame C++ New
Create AfxRegisteerWndClass

Create

LoadFrame Create
LoadFrame ID
IDR_MAINFRAME

CFrameWnd
CDocTemplate
CDocTemplate CRuntimeClass
File New MDI Windows New CRuntimeclass

CFrameWnd DECLARE_DYNCREATE
RUNTIME_CLASS

CFrameWnd Windows

· CFrameWnd Windows
CView::OnActivateView

· CFrameWnd
CWnd OnSetFocus OnHScroll OnVScroll

· MDI MDI
FMS_ADDTOTITLE

· CFrameWnd
CFrameWnd

· CFrameWnd
UPDATE_COMMAND_UI

· CFrameWnd

· CFrameWnd LoadFrame ID
SHIFT+F1

· CFrameWnd
DDE
ShellExecute

· CWinThread::m_pMainWnd
OnClose OnQueryEndSession

· WinHelp
WINHELP.EXE

C++ delete CWnd::DestroyWindow
CFrameWnd PostNcDestroy C++
OnClose DestroyWindow

CFrameWnd

CFrameWnd

Create

0

lpszClassName Windows
AfxRegisterJWndClass RegisterClassWindows
NULL CFrameWnd

lpszWindowName

dwStyle
FWS_ADDTOTITLE

rect rectDefault Windows

pParentWnd NULL

lpszMenuName ID
ID MAKEINTRESOURCE NULL

dwExStyle

pContext CCreateContext NULL

CFrameWnd CFrameWnd
Create Windows CFrameWnd Create

LoadFrame Create

CreateView

CWnd NULL

pContext

nID ID

CreateView CView

CreateView

MFC COLLECT CreateView Windows 95 3D

GetActiveDocument

CDocument NULL

CDocument

```
// Get the active MDI child window .
CMDIChildWnd* pChild = (CMDIChildWnd* )pFrame->GetActiveFrame( );

// or CMDIChildWnd* pChild = pFrame->MDIGetsActive( );

// Get the active view attached to the active MDI child windows.
CMyView * pView = (CMyView * ) pChild->GetActiveView( );
```

GetControlBar

ID

nID ID

ID

GetMessageString

nID ID

rMessage CString

ID nID

LoadFrame

nIDResource ID

dwDefaultStyle
FWS_ADDTOTITLE

pParentWnd

pContext CCreateContext NULL

CFrameWnd CFrameWnd
LoadFrame Windows CFrameWnd

Create LoadFrame

LoadFrame

pContext
LoadFrame pContext NULL

OnCreateClient

0

lpcs Windows CREATESTRUCT

pContext CCreateContext

OnCreate

pContext CView

CCreateContext
CCreateContext CCreateContext
CREATESTRUCT
CWnd PreCreateWindows

SetActiveView

pViewNew CView NULL

bNotify TRUE OnActiveView FALSE

SetActiveView

SetActiveView

pViewNew CView NULL

bNotify TRUE OnActiveView FALSE

SetActiveView

ShowOwnedWindows

bShow

CFrameWnd

CListView

CListView CListCtrl CListCtrl MFC
CView

CListView

CListView

GetListCtrl

CObject

CObject Microsoft CFile CObList
CObject

.

.

.

.

CObject CObject CObject
CObject

CObject

DECLARE_DYNAMIC IMPLEMENT_DYNAMIC

DECLARE_SERIAL IMPLEMENT_SERIAL
“ ”

Microsoft C++ CObject
“Visual C++ ” “CObject ” “
”

AssertValid

AssertValid AssertValid

AssertValid

AssertValid

AssertValid

AssertValid

AssertValid

“ ” “ ” “shallow check”

CObList::CObList CObject CAge

```
// example for CObject::AssertValid
void CAge::AssertValid() const
{
  CObject::AssertValid();
  ASSERT( m_years > 0 );
  ASSERT( m_years < 105 );
}
```

CObject

objectSrc CObject

CObject

IMPLEMENT_SERIAL

“ ” “Visual C++ ”

“CObject ”

C++

CObject

Dump

dc afxDump

CDumpContext

Dump

Dump Dump

IMPLEMENT_DYNAMIC IMPLEMENT_SERIAL CObject::Dump

Dump

Dump Microsoft

#ifdef _DEBUG/#endif

Dump CObject

CDumpContext (<<) Dump

Dump “ ”

CObList::CObList CObject CAge

```
// example for CObject::Dump
void CAge::Dump( CDumpContext &dc ) const
{
   CObject::Dump( dc );
   dc << "Age = " << m_years;
}
```

GetRuntimeClass

CRuntimeClass NULL

CObject CRuntimeClass

· LPCSTR m_lpszClassName ASCII

· int m_nObjectSize

· UINT m_wSchema -1 IMPLEMENT _SERIAL

· CObject* (PASCAL* m_pfnCreateObject)( )

NULL)

· CRuntimeClass* (PASCAL* m_pfn_GetBaseClass)( )

MFC AFXDLL CRuntimeClass

· CRuntimeClass* m_pBaseClass MFC CRuntimeClass

Visual C++ MFC
Visual C++ Editions

IMPLEMENT_DYNAMIC IMPLEMENT_SERIAL

CObList::CObList CObject CAge

```
// example for CObject::GetRuntimeClass
CAge a(21);
CRuntimeClass* prt = a.GetRuntimeClass();
ASSERT( strcmp( prt->m_lpszClassName, "CAge" ) == 0 );
```

IsKindOf

0

pClass CObject CRuntimeClass

pClass (1) (2)
DECLARE_DYNAMIC DECLARE_SERIAL

CArchive::ReadObject CArchive::WriteObject Serialize CArchive (<<) (>>)

“Visual C++ ” “ ”

ObList::CObList CObject CAge

```
// example for CObject::Serialize
void CAge::Serialize( CArchive& ar )
{
  CObject::Serialize( ar );
  if( ar.IsStoring() ) ar << m_years;
  else ar >> m_years;
}
```

CSocket

CSocket CAsyncsocket CAsyncsocket WindowsSockets API CAsyncsocket CSocket WindowsSockets API CSocket CSocketFile CArchive

CSocket CArchive Receive Send ReceiveFrom SendTo Accept CAsyncsocket CSocket WSAEWOULDBLOCK CancelBlockingCall WSAEINTR

CSocket Create Create CArchive

Connect Accept CSocketFile CSocketFile CSocket CArchive CArchive CSocketFile CArchive CSocketFile CSocket Win32 SDK “Windows 2 ” “Windows

Attach

hsocket

hsocket CSocket
m_socket

Win32 SDK “Windows Sockets ”

CancelBlockingCall

WSAEINTR

Connect Windows
socket

Accept

Closet Windows

Win32 SDK “Windows Sockets ”

Create

0 GetLastError

nSocketPort MFC
0

Create Windows
Create Bind

· SOCK_STREAM two-way
Internet TCP

· SOCK_DGRAM
Internet User Datagram Protocol UDP
CArchive

Accept CSocket
Accept
sock Create
Win32 SDK “Windows Sockets
”

CSocket

CSocket Create
Win32 SDK “Windows Sockets ”

FromHandle

CSocket hSocket CSocket
NULL

hSocket

CSocket
CSocket NULL

Win32 SDK “Windows Sockets ”

IsBlocking

0

Win32 SDK “Windows Sockets ”

OnMessagePending

0

Windows

Windows OnMessagePending

Win32 SDK “Windows Sockets ”

CSocketFile

CSocketFile Windows Sockets
CFile CSocketFile CSocket
—— —— CSocketFile CArchive MFC

CSocketFile
CSocket
CSocketFile CSocket

CSocketFile
CFile CSocketFile MFC
CSocketFile CFile MFC
CSocketFile CEditView
CEditView::SerializeRaw CSocketFile CArchive
CEditView CEditView::Serialize SerializeRaw
CSocketFile Seek

CSocketFile

pSocket CSocketFile

bArchiveCompatible CArchive
CSocketFile FALSE CFile
CSocketFile CArchive

CSocketFile

CSocketFile CArchive
CSocketFile bArchiveCompatible TRUE

bArchiveCompatible FALSE
“ ” CSocketFile “ ”

CSocketFile CArchive CFile
CFile

CSocketFile

CFile
CArchive::IsBufferEmpty

CString

CString

CString CString Basic
CString

CString TCHAR _UNICODE
TCHAR wchar_t 16 TCHAR char 8
UNICODE CString 16 UNICODE CString
8

_UNICODE CString MBCS
DBCS MBCS CString 8
MBCS

CString

· CString

· CString “ ” CString

· CString const char* LPCTSTR
· C-

CString

CString

MFC
CString CString::LockBuffer CString::UnlockBuffer
LockBuffer
-1 “ ”

UnlockBuffer

1

Compare

CString lpsz <0 CString lpsz >0

lpsz

_tcscmp CString _tcscmp TCHAR.H strcmp wcscmp _mbscmp

“Microsoft Visual C++ 6.0 ” strcmp wcscmp _mbscmp

CString::Compare

```
// CString::Compare
CString s1( "abc" );
CString s2( "abd" );
ASSERT( s1.Compare( s2 ) == -1 ); //  CString
ASSERT( s1.Compare( "abe" ) == -1 ); //  LPTSTR
```

CompareNoCase

CString lpsz <0 CString lpsz >0

_tcsicmp CString _tcsicmp TCHAR.H _stricmp

_wcsicmp _mbsicmp

“Microsoft Visual C++ 6.0

” _stricmp _wcsicmp _mbsicmp

CString::CompareNoCase

```
// CString::CompareNoCase
CString s1( "abc" );
CString s2( "ABD" );
ASSERT( s1.CompareNoCase( s2 ) == -1 ); //          CString
ASSERT( s1.Compare( "ABE" ) == -1 ); //      LPTSTR
```

CString

stringSrc CString CString

ch nRepeat

nRepeat ch

lpch nLength

nLength pch

psz CString

lpsz CString

CString

· CString(const unsigned char* psz) unsigned char CString

“Visual C++ ” “String:CString Exception Cleanup ”

String::CString

```
// CString::CString
CString s1; //
CString s2( "cat" ); //                C
CString s3 = s2; //
CString s4( s2 + " " + s3 ); //
CString s5( 'x' ); // s5 = "x"
CString s6( 'x', 6 ); // s6 = "xxxxxx"
CString s7((LPCSTR)ID_FILE_NEW); // s7 = "Create a new document"
CString city = "Philadelphia"; //
```

Delete

nIndex

nCount

nIndex nCount

```
//                    CString::Delete
str2 = "Hockey is best!";
printf( "Before: %s\n, (LPCTSTR) str2 );
int n = str2.Delete(6, 3);
printf( "After: %s\n*, (LPCTSTR) str2 );
ASSERT( n == str2.Getlength ( ) );
//
```

```
Before: Hockey is best!
After: Hockey best!
```

Empty

CString

“Visual C++ ” “ CString ”

CString::Empty

```
// CString::Empty
CString s( "abc" );
s.Empty();
ASSERT( s.GetLength( ) == 0 );
```

Find

CString

-1

ch

lpszSub

nStart 0 nStart 0 nStart

pstr

strchr strstr

```
//
```

```
// CString::Find( TCHAR ch )
CString s( "abcdef" );
ASSERT( s.Find( 'c' ) == 2 );
ASSERT( s.Find( "de" ) == 3 );

//
// CString::Find(TCHAR ch,int nStart)
CString str("The stars are aligned");
int n = str.Find('e',5);
ASSERT(n == 12)
```

FindOneOf

lpszCharSet

lpszCharSet

lpszCharSet

CString::FindOneOf

```
// CString::FindOneOf
CString s( "abcdef" );
ASSERT( s.FindOneOf( "xd" ) == 3 ); // 'd' is first match.
```

Format

lpszFormat

nFormatID

CString sprintf C-

CString

lpszFormat nFormatID

Format

CString str = "Some Data";

str.Format("%s%d",str, 123); // str

LPCTSTR

printf

“Microsofr Visual C++ 6.0 ” printf

“Microsofr Visual C++ 6.0 ” sprintf

GetAt

TCHAR

nIndex CString nIndex 0

GetLength Microsoft nIndex

CString GetAt

[] GetAt

CString::GetAt

```
// CString::GetAt
CString s( "abcdef" );
ASSERT( s.GetAt(2) == 'c' );
```

GetBuffer

LPTSTR

nMinBufLength

CString LPTSTR
const CString

GetBuffer CString
ReleaseBuffer

ReleaseBuffer GetBuffer CString
CString CString

CString

ReleaseBuffer
ReleaseBuffer -1 ReleaseBuffer strlen

CString::GetBuffer

```
// CString::GetBuffer
CString s( "abcd" );
#ifdef _DEBUG
   afxDump << "CString s" << s << "\n";
#endif
LPTSTR p = s.GetBuffer( 10 );
strcpy( p, "Hello" ); //              CString
s.ReleaseBuffer( );
#ifdef _DEBUG
   afxDump << "CString s " << s << "\n";
#endif
```

GetBufferSetLength

LPTSTR

nNewLength CString

Insert

nIndex

ch

pstr

nIndex nIndex

nIndex

ch pstr

```
//                                    CString::Insert
CString str("HockeyBest");
int n = str.Insert( 6, "is" );
ASSERT( n == str.GetLength( ) );
printf( "1: %s\n", ( LPCTSTR ) str );
n = str.Insert( 6, ' ' );
ASSERT( n == str.GetLength( ) );
printf ( *2: %s\n*, (LPCTSTR) STR );
n = str.Insert(555,'1');
ASSERT( n == str.GetLength ( ) );
printf ( "3: %s\n", ( LPCTSTR ) str );
//
1.Hockeyis Best
2.Hockey is Best
3.Hockey is Best!
```

IsEmpty

CString 0 0

CString

CString::IsEmpty

```
// CString::IsEmpty
CString s;
ASSERT( s.IsEmpty() );
```

Left

CString CString

nCount CString

CString nCount

nCount Left Basic LEFT$

MBCS nCount 8

CString::Left

```
// CString::Left
CString s( _T("abcdef") );
ASSERT( s.Left(2) == _T("ab") );
```

LoadString

0

nID Windows ID

nID Windows CString

CString::LoadString

```
// CString::LoadString
#define IDS_FILENOTFOUND 1
CString s;
if (! s.LoadString( IDS_FILENOTFOUND ))
{
  AfxMessageBox("Error Loading String: IDS_FILENOTFOUND");
  ...
}
```

LockBuffer

CString NULL

LockBuffer -1

-1 “ ”

.

.

LockBuffer UnlockBuffer 1

· “Win 32 SDK ” “ ”

· “Win 32 SDK ” “ ”

· “Win 32 SDK ” “ ”

```
CString s;s = "abc";
LPTSTR p = s.GetBuffer( 1024 );
strcpy(p, "abc"); //
ASSERT( s.GetLength() == 3 ); //          = 3
s.ReleaseBuffer(); //                   p
ASSERT( s.GetLength() == 3 ); //          3
```

Remove

ch

ch

// 'c':

```
CString str ("This is a test.");
int n = str.Remove( 't' );
ASSERT( n == 2 );
ASSERT( str == "This is a es." );
```

Replace

chOld chNew

chNew chOld

lpszOld lpszNew

lpszNew lpszOld

chNew

chOld lpszNew lpszOld

lpszNew lpszOld

```
//                    old      new
CString strZap( "C--" );
int n = strZap.Replace('-', '+' );
ASSERT( n == 2 );
ASSERT(strZap == "C++" );
//                    old     new
CString strBang("Everybody likes ice hockey");
n = strBang.Replace( "hockey", "golf" );
ASSERT( n == 1 )
n = strBang.Replace ( "likes" , "plays" );
ASSERT( n == 1 )
n = strBang.Replace( "ice", NULL );
ASSERT( n == 1 )
ASSERT( strBang == "Everybody plays golf" )
//
//                                                                    “ice”
```

ReverseFind

CString
-1

ch

CString
strrchr

```
// CString::ReverseFind
```

```
CString s( "abcabc" );
ASSERT( s.ReverseFind( 'b' ) == 4 );
```

Right

CString CString

nCount CString

CString nCount
nCount Right
Basic RIGHT$

MBCS nCount 8

CString::Right

```
// CString::Right
CString s( _T("abcdef") );
ASSERT( s.Right(2) == _T("ef") );
```

SetAt

nIndexCString nIndex 0
Getlength Microsoft nIndex Release

ch

CString SetAt
SetAt

TrimLeft

chTarget

lpszTargets

TrimLeft tab

TrimRight “Visual C++ ”
“ ”

TrimRight

chTarget

lpszTargets

TrimRight tab

```
CString strBefore;
CString strAfter;
  strBefore ="Hockey is Best!!!!" ;
  strAfter = strBefore;
  str.TrimRight('!' );
  printf ("Before: \"%s\"\n",(LPCTSTR) strBefore );
  printf ("After: \"%s\"\n",(LPCTSTR) strBefore );
```

```
strBefore ="Hockey is Best?!?!?!?!" ;
strAfter = strBefore;
str.TrimRight( *?!* );
printf ("Before: \"%s\"\n",(LPCTSTR) strBefore );
printf ( "After: \"%s\"\n",(LPCTSTR) strAfter );
                                    “Hockey is Best!!!!”        “Hockey is Best”
                                    “Hockey is Best?!?!?!?!”        “Hockey isBest”
                          “Visual C++                ”     “                ”
```

UnlockBuffer

LockBuffer UnlockBuffer

1

CString UnlockBuffer

CStringArray

CStringArray CString

CStringArray CObArray
CObArray CStringArray
CObject CString
CString CObject
LPCTSTR

CObject* CObArray::GetAt(int <nIndex>) const;

CString CStringArray::GetAt(int <nIndex>) const;

void SetAt(int <nIndex>, CObject* <newElement>)

void SetAt(int <nIndex>, LPCTSTR <newElement>);

CStringArray IMPLEMENT_SERIAL
CString Serialize

SetSize
SetSize

1

CString

CStringList

CStringList CString

CStringList CObList
CObList CStringList
CObject CString CString
CObject LPCTSTR

CObject*& CObList::GetHead() const;

CString& CStringList::GetHead() const;

POSITION AddHead(CObject* <newElement>);

POSITION AddHead(LPCTSTR <newElement>);

CStringList IMPLEMENT_SERIAL
CString Serialize

1

CStringList CString

nHour, nMin, nSec CTime

nDST daylight saving time

· nDST 0

· nDST 0 daylight saving time

· nDST 0
daylight saving time

wDosDate, wDosTime / CTime
MSDOS

sysTime / CTime SYSTEMTIME

fileTime / CTime FILETIME

CTime

· CTime( ); CTime CTime

· CTime(const CTime&); CTime CTime

· CTime(time_t); time_t CTime

· CTime(int, int, etc.); CTime

nYear 1970—3000

nMonth 1—12

nDay 1—31

nHour

nMin

nSec

* 1/18/2038 COleDateTime
UTC Microsoft

· CTime(WORD, WORD); MSDOS CTime
· CTime(const SYSTEMTIME&); SYSTEMTIME CTime

· CTime(const FILETIME&); FILETIME CTime
CTime FILETIME CTime
CFile::GetStatus FILETIME CTime

time_t “Microsoft Visual C++ 6.0 ”

“Win32 SDK ” SYSTEMTIME FILETIME
Win32 SDK MSDOS

```
// CTime::CTime
time_t osBinaryTime; // C ( <time.h> )
time( &osBinaryTime ); //
CTime time1; // CTime. (0 .)
CTime time2 = time1; //
CTime time3( osBinaryTime ); // C
CTimeCTime time4( 1999, 3, 19, 22, 15, 0 ); // 10:15PM March 19, 1999
```

Format

CString

pFormat printf %
CTime
strftime Format

· %D CTime

- %H
- %M
- %S
- %%

nFormatID ID

/ CTime

CTime

```
// CTime::Format     CTime::FormatGmt
CTime t( 1999, 3, 19, 22, 15, 0 ); // 10:15PM March 19, 1999
CString s = t.Format( "%A, %B %d, %Y" );
ASSERT( s == "Friday, March 19, 1999" );
```

FormatGmt

CString

pFormat printf strftime

nFormatID ID

CTime

UTC

GetAsSystemTime

0

timeDest SYSTEMTIME /
CTime

CTime Win32
SYSTEMTIME GetAsSystemTime timeDest
SYSTEMTIME wMilliseconds

GetCurrentTime

CTime

```
// CTime::GetCurrentTime
CTime t = CTime::GetCurrentTime();
```

GetDay

1--31 GetLocalTm
CTime

```
// CTime::GetDay, CTime::GetMonth,      CTime::GetYear
CTime t( 1999, 3, 19, 22, 15, 0 ); // 10:15PM March 19, 1999
ASSERT( t.GetDay() == 19 );
ASSERT( t.GetMonth() == 3 );
ASSERT( t.GetYear() == 1999 );
```

GetDayOfWeek

1 2 ... 7

GetLocalTm

GetHour

0 23 GetLocalTm
CTime

```
// CTime::GetHour, CTime::GetMinute,  CTime::GetSecond
CTime t( 1999, 3, 19, 22, 15, 0 ); // 10:15PM March 19, 1999
ASSERT( t.GetSecond() == 0 );
ASSERT( t.GetMinute() == 15 );
ASSERT( t.GetHour() == 22 );
```

GetLocalTm

struct tm struct tm TIME.H
GetGmtTm

ptm NULL
CTime

CTime struct tm
GetLocalTm

```
// CTime::GetLocalTm
CTime t( 1999, 3, 19, 22, 15, 0 ); // 10:15PM March 19, 1999
struct tm* osTime; //
osTime = t.GetLocalTm( NULL );
ASSERT( osTime->tm_mon == 2 ); // Note zero-based month!
```

e

0 59 GetLocalTm
CTime

n

1 12 1
CTime
GetDay

d

0 59 GetLocalTm
CTime

CTime time_t

GetTime

GetYear

1970 1 1 2038 1 18

GetLocalTm CTime

GetDay

CView

CView

CDocTemplate

MDI

CDocument::UpdateAllViews OnUpdate OnUpdate

CView OnDraw
OnDraw

CWnd::OnHScroll CWnd::OnVScroll
CView SCrollView

CScrollView CView

- CCtrlView /
- CDaoRecordView
- CEditView CEditView
- CFormView
- CListView /
- CRecordView
- CRichEditView /
- CScrollView
- CTreeView /

CView CPreviewView

CPreviewView

CView

“Visual C++ ” “ / ” “ ”

“Visual C++ ” “ 30”

CView

CView

OnInitialUpdate

GetDocument

NULL

IsSelected

0

pDocItem

FALSE

CDocItem OLE

OnActiveFrame

nstate

· WA_INACTIVE

· WA_ACTIVE

· WA_CLICKACTIVE

pFrameWnd

CFormView

OnActivateView

bActivate

pActivateView

pDeactiveView

bActivate

pActivateView

pDeactiveView ——

MDI

CFrameWnd::SetActiveView

CFrameWnd::GetActive-View

OnBeginPrinting

pDC

pInfo CPrintInfo

OnPreparePrinting

GDI

OnPrint GDI

GDI

Print

OnPreparePrinting

OnBeginPrinting

CDC

OnDropEx

OnEndPrinting

pDC

pInfo CPrintInfo

OnBeginPrinting.1 Tm<09961B442519526078 66.034 Td[<34EA44<02CA0CF11B4404

pInfo CPrintInfo

OnPrepareDC pInfo CPrintInfo
m_nCurPage OnDraw

.
OnDraw
.
OnDraw
GDI
GDI
OnBeginPrinting OnEndPrinting
. OnDraw OnDraw
pInfo m_rectDraw
OnPrint OnPrepareDC OnPrint
OnPrepareDC

OnPrint

```
void CMyView::OnPrint( CDC *pDC, CPrintInfo *pInfo )
{
// Print headers and/or footers, if desired.
// Find portion of document corresponding to pInfo->m_nCurPage
. OnDraw( pDC );
}
```

OnScroll

bDoScroll TRUE 0

bDoScroll FALSE bDoScroll TRUE

nScrollCode

- SB_BOTTOM
- SB_LINEDOWN
- SB_LINEUP

OLE

CWnd::OnHScroll CWnd::OnVScroll

OnUpdate

pSender NULL

lHint

pHint

CDocument::UpdateAllViews
OnInitialUpdate
WM_PAINT

lHint
pHint CObject
OnUpdate CObject::IsKindOf

OnUpdate
CWnd::InvalidateRect
WM_PAINT

lHint 0 pHint NULL

CWinApp

CWinApp Windows

CWinApp Windows
WinMain C++
WinMain CWinApp

CWinApp InitInstance

CWinApp
CWinApp

- AfxGetApp CWinApp
- AfxGetInstanceHandle
- AfxGetResourceHandle
- AfxGetAppName

CWinApp m_pszExename

AddDocTemplate

pTemplate CDocTemplate

RegisterShellFileTypes

```
BOOL CMyApp::InitInstance()
{
  // ...
  //                    MDI                    AppWizard
  CMultiDocTemplate* pDocTemplate;
  pDocTemplate = new CMultiDocTemplate(
     IDR_MYTYPE,
     RUNTIME_CLASS(CMyDoc),
     RUNTIME_CLASS(CMDIChildWnd), //          MDI
     RUNTIME_CLASS(CMyView)
    );

  AddDocTemplate(pDocTemplate);
  // ...
}
```

AddToRecentFileList

lpszPathName

lpszPathName MRU
LoadStdProfileSetting MRU

Save As

```
//                  c:\temp\test.doc      // File                  MRU

AfxGetApp()->AddToRecentFileList("c:\\temp\\test.doc");
```

CloseAllDocuments

bEndSession Windows TRUE FALSE

CloseAllDocuments HideApplication

CreatePrinterDC

0

dc

DC CreatePrinterDC

CDC CDC=::DeleteDC

CWinApp

lpszAppName null Windows NULL CWinApp AFX_IDS_APP_TITLE

CWinApp lpszAppName CWinApp CWinApp CWinApp WinMain

DoWaitCursor

nCode 1 0
-1

CWaitCursor CCmdTarget::BeginWaitCursor
CCmdTarget::EndWaitCursor CCmdTarget::RestoreWaitCursor
DoWaitCursor
DoWaitCursor
CwaitCursor

Enable3dControls

Enable3dControls Enable3dControlsStatic

CTL3D32.DLL TRUE FALSE
FALSE

InitInstance
CTL3D32.DLL
Enable3dControls Enable3dControlsStatic SetDialogBkColor

MFC DLL Enable3dControls MFC
Enable3dControlsStatic

Visual C++ MFC
“Visual C++”

MFC 3D

·CDialog

·CDialogBar

·CFormView

·CPropertyPage

·CPropertySheet

·CControlBar

·CToolBar

3D Enable3dControls
Enable3dControlsStatic 3D
CTL3D32 API

```
#ifdef _AFXDLL
  Enable3dControls( ); //        Enable3dControls
#else
  Enable3dControlsStatic( ); //        Enable3dControlsStatic
#endif
```

EnableShellOpen

InitInstance
Windows
RegisterShellFileTypes .REG

```
BOOL CMyApp::InitInstance()
{
  // ...
  CMultiDocTemplate* pDocTemplate;
  pDocTemplate = new CMultiDocTemplate(
    IDR_MYTYPE,
    RUNTIME_CLASS(CMyDoc),
    RUNTIME_CLASS(CMDIChildWnd), //          MDI
    RUNTIME_CLASS(CMyView));
  AddDocTemplate(pDocTemplate);
  //        MDI
  CMainFrame* pMainFrame = new CMainFrame;
  if (!pMainFrame->LoadFrame(IDR_MAINFRAME)) return FALSE;
  //
  //
```

```
    m_pMainWnd = pMainFrame;
    //                          /       DDE
    EnableShellOpen();
    RegisterShellFileTypes();
    // ...
    //                                  nCmdShow
    pMainFrame->ShowWindow(m_nCmdShow);
    pMainFrame->UpdateWindow();
    // ...
}
```

ExitInstance

0 0 WinMain

Run

Run

.INI

GetFirstDocTemplatePosition

POSITION NULL

GetNextDocTemplate

POSITION CDocTemplate

GetNextDocTemplate

CDocTemplate

pos POSITION GetNextDocTemplate GetFirstDoc Template

pos pos POSITION GetFirstDocTemplatePosition GetNextDocTemplate POSITION pos NULL

GetProfileString

.INI lpszDefault _MAX_PATH lpszDefault NULL

lpszSection null

lpszEntry null NULL

lpszDefault

.INI

·Windows NT

·Windows 3.X WIN.INI

·Windows 95 WIN.INI

```
CString strSection = "My Section";
CString strStringItem = "My String Item";
```

```
CString strIntItem = "My Int Item";
CWinApp* pApp = AfxGetApp();
pApp->WriteProfileString(strSection, strStringItem, "test");
CString strValue;
strValue = pApp->GetProfileString(strSection, strStringItem);
ASSERT(strValue == "test");
pApp->WriteProfileInt(strSection, strIntItem, 1234);
int nValue;
nValue = pApp->GetProfileInt(strSection, strIntItem, 0);
ASSERT(nValue == 1234);
```

HideApplication

InitInstance

0

Windows

```
{
  //
  //
  //
  SetDialogBkColor(); //
  LoadStdProfileSettings(); //                INI                MRU
  //
  //
  CSingleDocTemplate* pDocTemplate;
  pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CMyDoc),
    RUNTIME_CLASS(CMainFrame), // SDI
    RUNTIME_CLASS(CMyView)
   );
  AddDocTemplate(pDocTemplate);
  //
  OnFileNew();
  if (m_lpCmdLine[0] != '\0')
  {
    //
  }
  return TRUE;
}
```

LoadCursor

NULL

lpszResourceName null
CString

nIDResource ID

lpszResourceName nIDResource
LoadCursor

Windows LoadStandardCursor LoadOEMCursor

```
HCURSOR hCursor;
//                                    ID    IDC_MYCURSOR
hCursor = AfxGetApp()->LoadCursor(IDC_MYCURSOR);
HICON LoadIcon( LPCTSTR lpszResourceName ) const;
```

LoadIcon

NULL

lpszResourceName null
CString
nIDResource ID

lpszResourceName nIDResource

LoadStandardIcon LoadOEMIcon Windows

Win32 API LoadIcon SM_CXICON
SM_CYICON

LoadStandardCursor

NULL

lpszCursorName Windows IDC_
WINDOWS.H lpszCursorName

- IDC_ARROW
- IDC_IBEAM

- IDC_WAIT Windows
- IDC_CROSS
- IDC_UPARROW
- IDC_SIZE IDC_SIZEALL
- IDC_SIZEALL
- IDC_ICON IDC_ARROW
- IDC_SIZENWSE
- IDC_SIZENESW
- IDC_SIZEWE
- IDC_SIZENS

lpszCursorName Windows Windows
LoadStandardCursor LoadOEMCursor

```
HCURSOR hCursor;
//        Windows
hCursor = AfxGetApp()->LoadStandardCursor(IDC_UPARROW);
```

LoadStandardIcon

NULL

lpszIconName Windows
WINDOWS.H lpszIconName

- IDI_APPLICATION
- IDI_HAND
- IDI_QUESTION
- IDI_EXCLAMATION

· IDI_ASTERISK

lpszIconName Windows Windows
LoadStandardIcon LoadOEMIcon

OnContextHelp

CWinApp

ON_COMMAND(ID_CONTEXT_HELP, OnContextHelp)
SHIFT+F1

OnContextHelp

Windows WinHelp

OnFileNew

CWinApp

ON_COMMAND(ID_FILE_NEW, OnFileNew)

File New

22

```
//                AppWizard          File New  Open    Print Setup

BEGIN_MESSAGE_MAP(CMyApp, CWinApp)
//{{AFX_MSG_MAP(CMyApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
  // NOTE - the ClassWizard will add and remove mapping macros here.
  //    DO NOT EDIT what you see in these blocks of generated code!
```

```
//}}AFX_MSG_MAP
//
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
//
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()
//
// File New    Open     Print Setup
//         CWinApp
//         ClassWizard
//                          //{{AFX_MSG_MAP     //}}AFX_MSG_MAP
//
// CMyApp::OnFileNew              CMyApp::OnMyFileNew
BEGIN_MESSAGE_MAP(CMyApp, CWinApp)
//{{AFX_MSG_MAP(CMyApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
ON_COMMAND(ID_FILE_NEW, OnMyFileNew)
ON_COMMAND(ID_FILE_OPEN, OnMyFileOpen)
ON_COMMAND(ID_FILE_PRINT_SETUP, OnMyFilePrintSetup)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

OnFileOpen

CWinApp

ON_COMMAND(ID_FILE_OPEN, OnFileOpen)

File Open

22

```
//                        AppwWizard
// File New   Open     Print Setup
//
BEGIN_MESSAGE_MAP(CMyApp, CWinApp)
//{{AFX_MSG_MAP(CMyApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
  // NOTE - the ClassWizard will add and remove mapping macros here.
  //    DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
//
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
//
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()
//
// File New   Open     Print Setup
//
```

ON_COMMAND(ID_FILE_PRINT_SETUP, OnFilePrintSetup)

File Print

22

```
//                          AppWizard
// File New    Open     Print Setup
//
BEGIN_MESSAGE_MAP(CMyApp, CWinApp)
//{{AFX_MSG_MAP(CMyApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
// NOTE - the ClassWizard will add and remove mapping macros here.
//    DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
//
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
//
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()
//
// File New    Open     Print Setup
//         CWinApp
//         ClassWizard
//                          //{{AFX_MSG_MAP     //}}AFX_MSG_MAP
//
// CMyApp::OnFileNew              CMyApp::OnMyFileNew
BEGIN_MESSAGE_MAP(CMyApp, CWinApp)
//{{AFX_MSG_MAP(CMyApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
ON_COMMAND(ID_FILE_NEW, OnMyFileNew)
ON_COMMAND(ID_FILE_OPEN, OnMyFileOpen)
ON_COMMAND(ID_FILE_PRINT_SETUP, OnMyFilePrintSetup)
```

```
    //}}AFX_MSG_MAP
    END_MESSAGE_MAP()
```

OnHelp

CWinApp

ON_COMMAND(ID_HELP, OnHelp)

F1 F1

F1

WINHELP.EXE

ID WinHelp

OpenDocumentFile

CDocument NULL

lpszFileName

CDocument

```
    BOOL CMyApp::InitInstance()
    {
      // ...
      if (m_lpCmdLine[0] == '\0')
      {
        //
        OnFileNew();
```

```
        }
        else
        {
          //
          OpenDocumentFile(m_lpCmdLine);
        }
        // ...
      }
```

ParseCommandLine

rCmdInfo CCommandLineInfo

CCommandLineInfo::ParseParam

AppWizard MFC AppWizard Ccommand-LineInfo InitInstance ProcessShellCommand ParseCommandLine

1. InitInstance CCommandLineInfo ParseCommandLine
2. ParseCommandLine CCommandLineInfo::ParseParam

3.ParseParam CCommandLineInfo ProcessShellCommand

4.ProcessShellCommand

ParseCommandLine

CCommandLineInfo::m_nShellCommand

PreTranslateMessage

PreTranslateMessage

AppWizard MFC AppWizard CCommandLineInfo InitInstance ProcessShellCommand ParseCommandLine

1. InitInstance CCommandLineInfo ParseCommand-Line
2. ParseCommandLine CCommandLineInfo::ParseParam
3. ParseParam CCommandLineInfo ProcessShellCommand
4. ProcessShellCommand

CCommandLineInfo CCommandLineInfo::m_nShellCommand CCommandLineInfo

enum{ FileNew,FileOpen,FilePrint,FilePrintTo,FileDDE,};

CCommandLineInfo::m_nShellCommand

ProcessWndProcException

Windows Windows 0L 1L TRUE

e tr valign=top> pMsg MSG Windows

“Command failed” “Internal application error”

RegisterShellFileTypes

bCompat TRUE Print Print To DefaultIcon FALSE

Windows

AddDocTemplate RegisterShellFileTypes RegisterShellFile-Types EnableShellOpen

RegisterShellFileTypes CDocTemplate Windows .REG

CDocTemplate

Run

WinMain

Run Windows WM_QUIT Run OnIdle PreTranslateMessage Windows TranslateMessage Windows DispatchMessage

Run

RunAutomated

0

“/Automation” “-Automation”

OLE

“Visual C++ ” “ ”

RunEmbedded

0

“/Embedding” “-Embedding”

“Visual C++ ” “ ”

SaveAllModified

0

WM_QUERYENDSESSION

CDocument::SaveModified

SelectPrinter

hDevNames DEVNAMES

hDevMode DEVMODE

bFreeOld

Print

hDevMode hDevNames NULL SelectPrinter

SetDialogBkColor

clrCtlBk

clrCtlText

InitInstance

```
BOOL CMyApp::InitInstance()
{
  //
  SetDialogBkColor(); //
  LoadStdProfileSettings(); //          INI                MRU
  // ...
}
```

SetRegistryKey

lpszRegistryKey

nIDRegistryKey ID/

INI

m_pszRegistry Key CWinApp GetProfileInt GetProfileString WriteProfileInt WriteProfileString MRU

HKEY_CURRENT_USER\Software\\\\.

WriteProfileString

0

lpszSection null

lpszEntry null

lpszValue NULL lpszEntry

INI

- Windows NT
- Windows 3.x WIN.INI
- Windows 95 WIN.INI

```
CString strSection = "My Section";
CString strStringItem = "My String Item";
CString strIntItem = "My Int Item";
CWinApp* pApp = AfxGetApp();
pApp->WriteProfileString(strSection, strStringItem, "test");
CString strValue;
strValue = pApp->GetProfileString(strSection, strStringItem);
ASSERT(strValue == "test");
pApp->WriteProfileInt(strSection, strIntItem, 1234);
int nValue;
nValue = pApp->GetProfileInt(strSection, strIntItem, 0);
```

```
    ASSERT(nValue == 1234);
```

CWindowDC

CWindowDC CDC Windows GetWindowDC ReleaseDC CWindowDC CWnd

CWindowDC

pWnd

CWindowDC pWnd CWnd Windows GetWindowDC

Windows GetWindowDC Cresource-Exception Windows Windows 5

CWinThread

CWinThread CWinApp
CWinApp CWinThread CWinThread

CWinThread

CWinApp
CWinThread

CWinThread m_bAutoDelete
FALSE

MFC CWinThread
CWinThread

CWinThread MFC MFC
_beginthreadex MFC API

AfxBeginThread
AfxBeginThread CWinThread
CRuntimeClass AfxBeginThread
AfxBeginThread

AfxBeginThread CWinThread
AfxBeginThread CWinThread
CreateThread CWinThread
.

CreateThread

0

dwCreateFlags

· CREATE_SUSPENDED ResumeThread

· 0

nStackSize 0

lpSecurityAttrs SECURITY_ATTRIBUTES

AfxBeginThread

CreateThread

CWinThread

CWinThread CreateThread
AfxBeginThread AfxBeginThread CreateThread

ExitInstance

0 0
::GetExitCodeThread.

Run
InitInstance

Run

m_bAutoDelete CWinThread
ExitInstance
ExitInstance

GetMainWnd

K

OLE
CWinThread CWinApp::m_pActiveWnd

·THREAD_PRIORITY_BELOW_NORMAL

·THREAD_PRIORITY_LOWEST

·THREAD_PRIORITY_IDLE

“Win32 SDK ” 4

::SetThreadPriority

InitInstance

0

InitInstance InitInstance

AfxBeginThread

PostThreadMessage

0

message ID

PreTranslateMessage

PreTranslateMessage

pMsg MSG

Windows ::TranslateMessage ::DispatchMessage
Windows

ProcessMessageFilter

0

code lpMsg

lpMsg Windows MSG

Windows

ProcessWndProcException

WM_CREATE -1 0

e

pMsg MSG Windows

WM_CREATE

WM_PAINT WM_PAINT

ResumeThread

0xFFFFFFFF

1 1

SuspendThread

CREATE_SUSPENDED 1

0

Run

::GetExitCodeThread

Run Windows
WM_QUIT Run OnIdle
PreTranslateMessage
Windows ::TranslateMessage Windows
::DispatchMessage

Run

SetThreadPriority

0

nPriority

·THREAD_PRIORITY_TIME_CRITICAL

·THREAD_PRIORITY_HIGHEST

·THREAD_PRIORITY_ABOVE_NORMAL

·THREAD_PRIORITY_NORMAL

·THREAD_PRIORITY_BELOW_NORMAL

·THREAD_PRIORITY_LOWEST

·THREAD_PRIORITY_IDLE

“Win32 SDK ”
::SetThreadPriority

CreateThread

SuspendThread

0xFFFFFFFF

ResumeThread

CWnd

CWnd

CWnd Windows CWnd CWnd
Windows Windows
CWnd Create CWnd DestroyWindow
Windows

CWnd WndProc Windows
CWnd OnMessage OnMessage

CWnd Windows CWnd

CWnd CWnd
Create CWnd

CWnd DestroyWindow

CWnd
CFrameWnd,CMDIFrameWnd,CMDIChildWnd,CView CDialog,
CWnd CButton

BeginPaint

CWnd EndPaint

lpPaint PAINTSTRUCT

CWnd PAINTSTRUCT

RECT

Invalidate InvalidateRect InvalidateRgn

BeginPaint WM_ONERASEBKGND

WM_PAINT BeginPaint

BeginPaint EndPaint

BeginPaint

BindDefaultProperty

dwDispID DISPID

vtProp -- VT_BSTR VT_VARIANT

szFieldName

pDSCWnd DCS

ID GetDlgItem

DataSource UserName Password SQL

CWnd

BindDefaultProperty

```
BOOL CMyDlg::OnInitDialog()
{
    ...
    CWnd* pDSC = GetDlgItem(IDC_REMOTEDATACONTROL);
    CWnd* pList = GetDlgItem(IDC_DBLISTBOX);
    pList->BindDefaultProperty(0x2, VT_BSTR, _T("CourseID"), pDSC);
    CWnd* pEdit = GetDlgItem(IDC_MASKEDBOX);
    pEdit->BindDefaultProperty(0x16, VT_BSTR, _T("InstuctorID"), pDSC);
    ...
```

```
    return TRUE;
}
```

BindProperty

dwDispID DISPID

pWndDSC DCS

ID GetDlgItem

MFC

BindProperty

```
BOOL CMyDlg::OnInitDialog()
{
    ...
    CWnd* pDSC = GetDlgItem(IDC_REMOTEDATACONTROL);
    CWnd* pList= GetDlgItem(IDC_DBLISTBOX);
    pList.BindProperty(0x9, pDSC);
    ...
    return TRUE;
}
```

CancelToolTips

bKeys TRUE

FALSE

: CWnd::Enable

ToolTips

CenterWindow

pAlternateOwner

CDialog::OnInitDialog

pAlternateOwner
pAlternateOwner CWnd::GetDesktopWindow

Create

0

lpszClassName null Windows
WNDCLASS AfxRegisterWndClass
NULL CWnd

lpszWindowName null

dwStyle WS_POPUP
CWnd::CreateEx

rect pParentWnd

pParentWnd

nID ID

pContext

Windows CWnd

nID ID

pPersist CFile NULL
NULL
CFile
CFile
CreateControl

bStorage pPersist IStorage IStream
pPersist bStorage TRUE pPersist
bStorage FALSE FALSE

bstrLicKey
NULL

clsid ID

OLE MFC CWnd
CreateControl CWnd::Create CWnd::Create CWnd CreateControl
OLE

CreateControl Windows dwStyle

· WS_VISIBLE

· WS_DISABLED
Enable

· WS_BORDER BorderStyle

· WS_GROUP
WS_GROUP
WS_GROUP

· WS_TABSTOP TAB TAB
WS_TABSTOP

CWnd

CWnd CreateEx Create Windows

Default

DefWindowProc

message Windows

wParam

lParam

DeleteTempMap

CWinApp FromHandle
CWnd

DestroyWindow

0

CWnd Windows DestroyWindow

CWnd
WM_DESTROY
WM_NCDESTROY CWnd

DestroyWindow DestroyWindow
ClassWizard CWnd CWnd
DestroyWindow MFC
DestroyWindow

CView DestroyWindow MFC
CFrameWnd DestroyWindow
DestroyWindow

DestroyWindow

DestroyWindow CDialog::Create
CWnd WS_EX_NOPARENTNOTIFY
WM_PARENTNOTIFY

EndPaint

lpPaint PAINTSTRUCT BeginPaint

BeginPaint
EndPaint

BeginPaint EndPaint

FindWindow

NULL

CWnd*

lpszClassName null WNDCLASS
lpClassName NULL

FlashWindow

bInvert

GetActiveWindow

NULL

SetActiveWindow

GetCurrentMessage

MSG

OnMessage

GetDC

CWnd NULL

CWnd GetDC

GetDC

GDI

ReleaseDC

WNDCLASS CS_CLASSDC CS_OWNDC CS_PARENTDC GetDC CWnd

GetDesktopWindow

Windows

Windows

GetFocus

NULL

CWnd

GetIcon

NULL

bBigIcon TRUE 32×32 FALSE 16×16

32×32 16×16 bBigIcon

GetNextWindow

nFlag
GW_HWNDNEXT CWnd
GW_HWNDPREV

CWnd CWnd

GetParent

NULL

GetParent
GetParentOwner WS_CHILD
GetParent
GetParentOwner

DWORD GetStyle( ) const;

GetSystemMenu

bRevert FALSE TRUE

bRevert bRevert FALSE GetSystemMenu bRevert TRUE GetSystemMenu

GetSystemMenu

GetSystemMenu CMenu::AppendMenu CMenu::InsertMenu CMenu::ModifyMenu

ID SC_CLOSE SC_MOVE SC_SIZE WM_SYSCOMMAND 0xF000 ID F000 ID

Windows CWnd WM_INITMENU

GetTopWindow

CWnd NULL

CWnd CWnd NULL

GetWindow

NULL

nCmd CWnd

· GW_CHILD CWnd

· GW_HWNDFIRST CWnd

· GW_HWNDLAST CWnd

· GW_HWNDNEXT

· GW_HWNDPREV

· GW_OWNER CWnd

GetWindowDC

NULL

GetWindowDC

ReleaseDC

CWnd CWnd

GetWindowDC CWnd

Windows GetSystemMetrics

ReleaseDC

GetWindowText

CWnd

0

lpszStringBuf

nIDEvent SetTimer

SetTimer nIDEvent
WM_TIMER

void MoveWindow(int x, int y, int nWidth, int nHeight, BOOL bRepaint = TRUE);

MoveWindow

x CWnd

y CWnd

nWidth CWnd

nHeight CWnd

bRepaint CWnd TRUE CWnd OnPaint
WM_PAINT FALSE
CWnd
FALSE CWnd

lpRect CRect RECT

CWnd x y

MoveWindow WM_GETMINMAXINFO CWnd
MoveWindow
WM_GETMINMAXINFO

OnActivate

nState CWnd

· WA_INACTIVE

· WA_ACTIVE

· WA_CLICKACTIVE

pWndOther CWnd NULL

bMinimized CWnd TRUE
TRUE CWnd

CWnd
OnActivate OnActivate

CWnd OnMouseActivate

Windows

OnActivateApp

bActive CWnd TRUE CWnd
FALSE CWnd

hTask bActive TRUE
CWnd bActive FALSE CWnd

Windows

OnCancelMode

CWnd CWnd
OnCancelMode CWnd

Windows ReleaseCapture

OnChar

nChar

nRepCnt

nFlags

0--15

16--23 OEM

24 101 102 ALT CTRL
1 0

25--28 Windows

29 ALT 1 0

30 1
0

31 1 0

OnKeyUp
OnKeyDown OnChar

OnChar nFlags
NFlags OnChar OnKeyUp
OnKeyDown

IBM 101 102 ALT CTRL
INS DEL HOME END PAGE UP PAGE DOWN
/ ENTER nFlags

Windows

OnChildNotify

0

message Windows

wParam wparam

lParam lparam

pLResult
NULL

0

lpCreateStruct CREATESTRUCT CWnd

Create CreateEx Windows
CWnd
OnCreate Create CreateEx

CREATESTRUCT

Windows

OnDestroy

CWnd OnDestroy CWnd

CWnd OnDestroy CWnd
OnDestroy OnDestroy

CWnd SetClipboardViewer
CWnd OnDestroy ChangeClipboardChain

OnEnable

bEnable CWnd CWnd
TRUE CWnd FALSE

IBM 101 102 ALT CTRL
INS DEL HOME END PAGE UP PAGE DOWN
/ ENTER nFlags

Windows

OnKeyUp

nChar

nRepCnt

nFlags

0-7 OEM

8 1

9-10

11-12 Windows

13 ALT 1 0

14 1 0

15 1 0

WM_KEYDOWN 15 1 13 0

ALT

CWnd

IBM 101 102 ALT CTRL
INS DEL HOME END PAGE UP PAGE DOWN
/ ENTER nFlags

Windows

OnKillFocus

pNewWnd NULL

CWnd

Windows

OnLButtonDblClk

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

CS_DBLCLKS WNDCLASS OnLButtonDlbClk

point x y

CS_DBLCLKS WNDCLASS OnLButtonDlbClk

Windows OnLButtonDblClk
WM_LBUTTONDOWN WM_LBUTTONUP WM_LBUTTONDBLCLK
WM_LBUTTONUP

Windows

OnLButtonDown

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

Windows

afx_msg void OnLButtonUp(UINT nFlags, CPoint point);

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

Windows

OnMButtonDblClk

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

CS_DBLCLKS WNDCLASS OnMButtonDlbClk

Windows OnMButtonDblClk
WM_MBUTTONDOWN WM_MBUTTONUP WM_MBUTTONDBLCLK
WM_MBUTTONUP

Windows

OnMButtonDown

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

Windows

OnMButtonUp

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

Windows

OnMouseActivate;

CWnd

- MA_ACTIVATE CWnd
- MA_NOACTIVATE CWnd
- MA_ACTIVATEANDEAT CWnd
- MA_NOACTIVATEANDEAT CWnd

pDesktopWnd

nHitTest

message

TRUE

OnNcHitTest

Windows

OnMouseMove

nFlags

MK_CONTROL CTRL

MK_LBUTTON

MK_MBUTTON

MK_RBUTTON

MK_SHIFT SHIFT

point x y

WM_MOUSEMOVE

Windows

OnMouseWheel

0

nFlags

· MK_CONTROL CTRL

· MK_LBUTTON

· MK_MBUTTON

· MK_RBUTTON

· MK_SHIFT SHIFT

zDelta zDelta WHEEL_DELTA 120

pt x y

OnMouseWheel WM_MOUSEWHEEL Windows

Win32 DefWindowProc

zDelta WHEEL_DELTA 120
delta
delta 120

delta WHEEL_DELTA delta
delta
WHEEL_DELTA

OnMouseWheel Windows NT 4.0 Windows 95 Windows NT 3.51
OnRegisteredMouseWheel

OnMove

x x

y y

CWnd

Windows

OnMoving

nSide

lpRect CRect RECT

CWnd

Windows

OnNotify

0

wParam wParam 0

lParam NMHDR

NMHDR

pResult LRESULT

OnNotify
WM_NOTIFY OnNotify

WM_NOTIFY “TN061 ON_NOTIFY WM_NOTIFY ”
“TN060 Windows ” “TN062 Windows ”
Visual C++

OnPaint

Windows
WM_PAINT UpdateWindow RedrawWindow
RDW_INTERNALPAINT RedrawWindow

GetUpdateRect

GetUpdateRect 0 BeginPaint EndPaint

WM_PAINT

WM_PAINT

RDW_INTERNALPAINT RedrawWindow

Windows WM_PAINT UpdateWindow

WM_PAINT WM_PAINT

RDW_INTERNALPAINT RedrawWindow

/ CView::OnDraw

WM_PAINT Win32 SDK

·“WM_PAINT ”

·“ WM_PAINT ”

OnRButtonDblClk

nFlags

· MK_CONTROL CTRL

· MK_LBUTTON

· MK_MBUTTON

· MK_RBUTTON

· MK_SHIFT SHIFT

point x y

CS_DBLCLKS WNDCLASS OnRButtonDlbClk

Windows OnRButtonDblClk

WM_RBUTTONDOWN WM_RBUTTONUP WM_RBUTTONDBLCLK

WM_RBUTTONUP

Windows

OnRButtonDown

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

Windows

OnRButtonUp

nFlags

- MK_CONTROL CTRL
- MK_LBUTTON
- MK_MBUTTON
- MK_RBUTTON
- MK_SHIFT SHIFT

point x y

Windows

OnSetCursor

0

pWnd

nHitTest

message

CWnd

OnSetCursor TRUE

nHitTest HTERROR MessageBeep

CWnd 0

Windows

OnSetFocus

pOldWnd CWnd NULL

CWnd

Windows

OnShowWindow

bShow TRUE
FALSE

nStatus ShowWindow
0 nStatus

· SW_PARENTCLOSING

· SW_PARENTOPENING f

CWnd ShowWindow

Windows

OnSize

nType

· SIZE_MAXIMIZED

· SIZE_MINIMIZED

· SIZE_RESTORED SIZE_MINIMIZED SIZE_MAXIMIZED

· SIZE_MAXHIDE

· SIZE_MAXSHOW

cx

cy

OnSize SetScrollPos MoveWindow SetScrollPos MoveWindow bRedraw CWnd Windows

OnTimer

nIDEvent

SetTimer

Windows DispatchMessage WM_TIMER

Windows

PostMessage

0

message

wParam

lParam

Windows GetMessage PeekMessage

Windows PostMessage

RedrawWindow

0

lpRectUpdate RECT prgnUpdate

prgnUpdate prgnUpdate lpRectUpdate NULL

flags

RDW_ERASE WM_ERASEBKGND
RDW_INVALIDATE RDW_ERASE

RDW_FRAME
WM_NCPAINT RDW_INVALIDATE RDW_FRAME

RDW_INTERNALPAINT WM_PAINT

RDW_INVALIDATE lpRectUpdate prgnUpdate NULL
NULL

RDW_NOERASE WM_ERASEBKGND

RDW_NOFRAME WM_NCPAINT
RDW_VALIDATE RDW_NOCHILDREN

RDW_NOINTERNALPAINT WM_PAINT
WM_PAINT

RDW_VALIDATE lpRectUpdate prgnUpdate NULL
NULL WM_PAINT
RedrawWindow

RDW_ERASENOW
RDW_ALLCHILDREN RDW_NOCHILDREN WM_NCPAINT
WM_ERASEBKGND WM_PAINT

RDW_UPDATENOW
RDW_ALLCHILDREN RDW_NOCHILDREN WM_NCPAINT
WM_ERASEBKGND WM_PAINT

RedrawWindow
WS_CLIPCHILDREN WS_CLIPCHILDREN
WS_CLIPCHILDREN
WS_CLIPCHILDREN RedrawWindow

RDW_ALLCHILDREN

RDW_NOCHILDREN

RedrawWindow
WM_PAINT CWnd::ValidateRgn
CWnd::InvalidateRgn CWnd::UpdateWindow ::RedrawWindow

ReleaseDC

0

pDC

ReleaseDC

GetWindowDC GetDC ReleaseDC

SendMessage

message

wParam

lParam

SendMessage
PostMessage

SendNotifyMessage

0

message

wParam

lParam

SendNotifyMessage

SendNotifyMessage

SetActiveWindow

CWnd

SetActiveWindow
Windows

SetCapture

NULL

CWnd

CWnd ReleaseCapture

SetFocus

NULL

SetFocus WM_KILLFOCUS
WM_SETFOCUS

SetTimer

KillTimer

0

nIDEvent

nElapse

lpfnTimer TimerProc
WM_TIMER NULL WM_TIMER
CWnd

WM_TIMER
TimerProc

lpfnTimer TimerProc

```
void CALLBACK EXPORT TimerProc(
  HWND hWnd,      //        SetTimer     CWnd
  UINT nMsg,      // WM_TIMER
  UINT nIDEvent    //
  DWORD dwTime    //
  );
```

SetTimer

SetWindowPos

0

pWndInsertAfter Z CWnd CWnd
CWnd

wndBottom Z CWnd

wndTop Z

wndTopMost

wndNoTopMost

x

y

cx

cy

nFlags
SWP_DRAWFRAME

SWP_FRAMECHANGED WM_NCCALCSIZE
WM_NCCALCSIZE

SWP_HIDEWINDOW

SWP_NOACTIVATE
pWndInsertAfter
SWP_NOCOPYBITS

SWP_NOMOVE x y
SWP_NOOWNERZORDER Z

SWP_NOREDRAW

SWP_NOREPOSITION SWP_NOOWNERZORDER

SWP_NOSENDCHANGING WM_WINDOWPOSCHANGING

SWP_NOSIZE cx cy

SWP_NOZORDER pWndInsertAfter

SWP_SHOWWINDOW

Z

Z Z

Z pWndInsertAfter
&wndTopMost SWP_NOZORDER Z

Z &wndBottom

SWP_NOACTIVE SWP_NOZORDER
Z pWndInsertAfter

· pWndInsertAfter &wndTopMost &wndNoTopMost
·

Z

Z

Windows 3.1 Z
WS_EX_TOPMOST

WinHelp Always On Top

SetWindowPos pWndInsertAfter
&wndTopMost WS_EX_TOPMOST

Z WS_EX_TOPMOST &wndTopMost

WS_EX_TOPMOST

SetWindowPos pWndInsertAfter &wndBottom CWnd
WS_EX_BOTTOM
Z

SetWindowText

lpszString CString null

WM_SETTEXT

ShowWindow

CWnd 0

nCmdShow CWnd

SW_HIDE

SW_MINIMIZE

SW_RESTORE Windows

SW_SHOW

SW_SHOWMAXIMIZED

SW_SHOWMINIMIZED

SW_SHOWMINNOACTIVE

SW_SHOWNA

SW_SHOWNOACTIVATE

SW_SHOWNORMAL Windows

CWinApp::m_nCmdShow ShowWindow
ShowWindow CWinApp::m_nCmdShow

UpdateData

0 bSaveAndValidate TRUE

bSaveAndValidate FALSE TRUE

CDialog::OnInitDialog
UpdateData bSaveAndValidate FALSE
CDialog::OnOK bSaveAndValidate TRUE
Cancel

WindowProc

message Windows

wParam

lParam

CWnd Windows WindowProc