

Reading

- Petzold Chapter 5
- Online Documentation for CDC

Practice Assignment

- Practice 2: Intro to Graphics

GDI

- Graphics Device Interface
- Device Independent Graphics Interface
 - doesn't care what video card
 - could be a printer

The Device Context (DC)

- All Graphics Output through DC
 - provides access to device
 - stores context info
 - Pen, Brush, Font, ...
 - current pen location
 - background color
 - text color
 - DC encapsulated by MFC CDC class

CDC Functions (from Petzold)

- create or destroy a DC
- get information from a DC
- draw something
- get or set attributes
- deal with GDI objects

Drawing from the Framework

- drawing in OnDraw
 - called automatically when window exposed
 - and when document updated

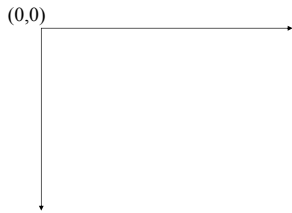
```
void OnDraw(CDC *pDC);
```

Simple Example

```
pDC->MoveTo(10,10);  
pDC->LineTo(100,100);
```

Coordinate System

- Default System



Utility Classes used by DCs

CPoint, CRect, CSize

- corresponding POINT, RECT, SIZE

```
typedef struct tagPoint {  
    LONG x;  
    LONG y;  
} POINT
```

Utility Classes (cont)

- MFC classes have nice properties
 - InflateRect, DeflateRect, PtInRect
 - IntersectRect, UnionRect
- can be used in place of Win32 structs
- CRect can also be used for LPRECT

Utility Classes (cont)

- COLORREF type
- RGB macro

Working with GDI Objects

- CBrush
- CPen
- CFont

Working with GDI Objects

- Generally, GDI Objects must be selected in to DC
- Rectangle uses current Pen and Brush
- FillRect takes a Brush

GDI Rules

- Always restore previously selected object
- Always cleanup after yourself

Sandwich Model

```
void CGraphicsTestView::OnDraw(CDC* pDC)
{
    ...
    CPen newPen(PS_SOLID,4,RGB(255,0,0));
    CPen *pOldPen = pDC->SelectObject(&newPen);

    pDC->MoveTo(10,10);
    pDC->LineTo(100,100);

    pDC->SelectObject(pOldPen);
}
```

Pre-defined GDI Objects

SelectStockObject

- BLACK_BRUSH, GREY_BRUSH
- SYSTEM_FIXED_FONT
- must convert type returned

Using a Brush

```
... CPen redPen(PS_SOLID,5,RGB(255,0,0));
     CPen *pOldPen = pDC->SelectObject(&redPen);

     CBrush greenBrush(HS_CROSS,RGB(0,255,0));
     CBrush *pOldBrush = pDC->SelectObject(&greenBrush);

     CRect drawRect(20,20,100,100);

     pDC->Rectangle(drawRect);

     pDC->SelectObject(pOldBrush);
     pDC->SelectObject(pOldPen);
```

Text Output

```
... CFont *pOldFont = (CFont *)
    pDC->SelectStockObject(ANSI_FIXED_FONT);

pDC->TextOut(120,50,"Hello",5);

pDC->SelectObject(pOldFont);
```

Getting Information

- Previous functions output information
- The following functions retrieve information

Getting the Window Size

- GetClientRect

Example:

```
CRect cltRect;
GetClientRect(cltRect);

CString drawString("In the Center");
pDC->DrawText(drawString,cltRect,
    DT_CENTER | DT_VCENTER | DT_SINGLELINE);
```

Device Capabilities

GetDeviceCaps (on CDC)

- TECHNOLOGY
- HORZSIZE, VERTSIZE
- HORZRES, VERTRES

System Metrics

::GetSystemMetrics

- information about system including
 - size of icons
 - thickness of frame windows
 - size of scroll bars
- Win32 function

OnDraw Details

OnDraw not a direct response to a Window's Message

- initiated by WM_PAINT Message
- normally need to call
 - BeginPaint (sets up DC and clipping)
 - EndPaint

Advanced DCs

- accessing DC outside of WM_PAINT context

Example:

- drawing directly in response to mouse movements

CClientDC

- Constructor takes a CWnd*
- Encapsulates
 - GetDC
 - ReleaseDC

CWindowDC

- Constructor takes a CWnd*
- Encapsulates
 - GetWindowDC
 - ReleaseDC