Overview

This chapter looks at simple graphics. There is a coverage of:

- Device contexts.
- Bitmaps.
- Brushes.
- Pens.
- Colour.
- Coordinate systems.

Graphics using device contexts

This chapter looks at simple graph plotting using the MFC library.

Graphic objects

Windows provides a variety of drawing tools to use in device contexts. We have already seen how to use brushes to fill interiors and fonts to draw text. In this chapter we will use pens to draw lines. MFC provides graphic-object classes equivalent to the drawing tools in windows. The table below shows the available classes and the equivalent windows GDI handle types.
Classes for windows GDI objects

### Table 15.1 GDI objects and classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Windows handle type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPen</td>
<td>HPEN</td>
</tr>
<tr>
<td>CBrush</td>
<td>HBRUSH</td>
</tr>
<tr>
<td>CFont</td>
<td>HFONT</td>
</tr>
<tr>
<td>CBitmap</td>
<td>HBITMAP</td>
</tr>
<tr>
<td>CPalette</td>
<td>HPALETTE</td>
</tr>
<tr>
<td>CRgn</td>
<td>HRGN</td>
</tr>
</tbody>
</table>

Each graphic-object class in the class library has a constructor that allows you to create graphic objects of that class, which you must then initialise with the appropriate function.

The field of graphics is quite extensive. I would recommend the following as a good place to start. There are several versions available.


Process of creating a graphic object in a device context

The following four steps are typically used when you need a graphic object for a drawing operation:

- Define a graphic object on the stack frame. Construct and initialise the object in two separate stages. The constructor creates the object and an initialization function initialised it. Two-stage construction is preferred to one stage as it is safer.
- Select the object into the current device context, saving the old graphic object that was selected before.
- When done with the current graphic object, select the old graphic object back into the device context to restore its state.
- Allow the frame-allocated graphic object to be deleted automatically when the scope is exited.

Obviously if you use an object repeatedly, you can allocate it once and select it into a device context each time it is needed. Be sure to delete such an object when you no longer need it.