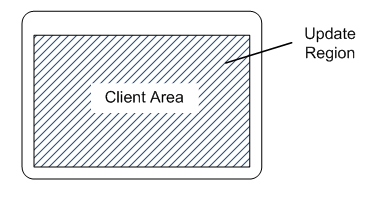
**РИСОВАНИЕ**

(Дополнительный лекционный материал)

Программа может инициировать рисование, чтобы обновить внешний вид окна. В других случаях это делает операционная система (например, при растяжке окна). Когда это происходит, операционная система отправляет окну сообщение WM\_PAINT. Часть окна, которая должна быть нарисована, называется областью обновления.

При первом отображении окна вся клиентская область окна должна быть закрашена. Следовательно, вы всегда будете получать по крайней мере одно сообщение WM\_PAINT при отображении окна.



(Update region – область обновления)

В Win API имеется функия  **UpdateWindow(),** котораяобновляет содержимое окна, посылая сообщение **WM\_PAINT** непосредственно в оконную процедуру.

BOOL UpdateWindow

(

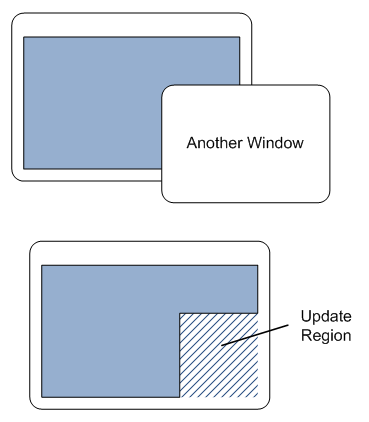
HWND hWnd // окно

);

При успешном выполнении возвращается ненулевое значение. То же самое можно сделать с помощью **InvalidateRect()**. Но функция **InvalidateRect() посылает** сообщение перерисовки непосредственно в оконную процедуру минуя очередь сообщений, то есть обновление происходит сразу без задержки,

Программа ответственна за заливку клиентской области. Окружающая рамка, включая строку заголовка, автоматически окрашивается операционной системой. После того, как вы закончите рисовать клиентскую область, следует очистить область обновления.

Теперь предположим, что пользователь перемещает другое окно, чтобы оно закрывало часть вашего окна. Когда затемненная часть снова становится видимой, эта часть добавляется в область обновления, и ваше окно получает другое сообщение WM\_PAINT.



Область обновления также изменяется, если пользователь растягивает окно.

Следующий простой код показывает реакцию на сообщение WM\_PAINT. Дело сводится к рисованию закрашенного прямоугольника (FillRect). Закрашивание выполняется кистью (BRUSH)

C++

switch (uMsg)

{

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hwnd, &ps);

// All painting occurs here, between BeginPaint and EndPaint.

FillRect(hdc, &ps.rcPaint, (HBRUSH) (COLOR\_WINDOW+1));

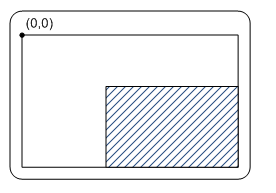
EndPaint(hwnd, &ps);

}

return 0;

}

Запустите операцию рисования, вызвав функцию BeginPaint. Эта функция заполняет структуру PAINTSTRUCT информацией о запросе на перерисовку. Текущий регион обновления указан в переменной rcPaint структуры PAINTSTRUCT. Этот регион обновления определяется относительно клиентской области:



В вашем коде рисования есть два основных варианта:

• Нарисовать всю клиентскую область, независимо от размера области обновления. Все, что выходит за пределы области обновления, обрезается.

• Рисовать только часть окна внутри области обновления.

Если вы всегда рисуете всю клиентскую область, код будет проще. Однако если у вас сложная логика рисования, может быть эффективнее пропустить области за пределами области обновления.

Следующая строка кода заполняет область обновления одним цветом, используя системный цвет фона окна (COLOR\_WINDOW). Фактический цвет, указанный в COLOR\_WINDOW, зависит от текущей цветовой схемы пользователя.

C++

FillRect(hdc, &ps.rcPaint, (HBRUSH) (COLOR\_WINDOW+1));

Функция FillRect является частью графического интерфейса устройства (GDI), который очень долго поддерживал графику Windows. В Windows 7 Microsoft представила новый графический движок под названием Direct2D, который поддерживает высокопроизводительные графические операции, такие как аппаратное ускорение. Direct2D также доступен для Windows Server 2008 и последующих версий (GDI по-прежнему полностью поддерживается.)

После того, как вы закончите рисование, вызовите функцию EndPaint. Эта функция очищает область обновления, которая сообщает Windows, что окно завершило рисование самостоятельно.

Элемент canvas позволяет отображать графику на зависимом от разрешения растровом холсте. Для рисования на холсте используются «контексты», такие как контекст Canvas 2D, указанный в спецификации HTML-контекста Canvas 2D. В Windows Internet Explorer 9 появилась поддержка элемента canvas, использующего API-интерфейс рисования 2D Canvas в качестве контекста. (В Internet Explorer 9 контекст Canvas 2D представлен объектом CanvasRenderingContext2D или интерфейсом ICanvasRenderingContext2D.) Как и вся графика в Internet Explorer 9, Canvas аппаратно ускоряется с помощью Windows и графического процессора.

Canvas позволяет рисовать сценарии, которые включают в себя прямоугольники, контуры, линии, заливки, дуги, кривые Безье и квадратные кривые. Кроме того, элемент canvas в Internet Explorer 9 поддерживает атрибуты width и height. (Значения по умолчанию для ширины и высоты составляют 300 и 150 пикселей соответственно, а цвет по

Сanvas позволяет использовать следующие фигуры рисования, которые включают:

* [rectangles](http://go.microsoft.com/fwlink/p/?linkid=197021)
* [lines](http://go.microsoft.com/fwlink/p/?linkid=197022)
* [fills](http://go.microsoft.com/fwlink/p/?linkid=197023)
* [arcs](http://go.microsoft.com/fwlink/p/?linkid=197024)
* [shadows](http://go.microsoft.com/fwlink/p/?linkid=197026)
* [Bézier curves](http://go.microsoft.com/fwlink/p/?linkid=197027)
* [quadratic curves](http://go.microsoft.com/fwlink/p/?linkid=197028)
* [images](http://go.microsoft.com/fwlink/p/?linkid=197030)
* [video](http://go.microsoft.com/fwlink/p/?linkid=197031)

Функции API для холста реализованы в броузере IE.

Так, Internet Explorer 9 и более поздние версии поддерживают следующие API-интерфейсы 2D-контекста Canvas (члены, предоставляемые объектом CanvasRenderingContext2D):

• свойство canvas (обратная ссылка на объект canvas)

• методы состояния (сохранить, восстановить)

• методы преобразования (масштабирование, поворот, преобразование, преобразование, setTransform)

• свойства компоновки (globalAlpha, globalCompositeOperation)

• свойства цвета и стиля (strokeStyle, fillStyle)

• объект и методы CanvasGradient (addColorStop, createLinearGradient, createRadialGradient)

• объект и метод CanvasPattern (createPattern)

• атрибуты линии (lineWidth, lineCap, lineJoin, miterLimit)

• Internet Explorer 11 добавляет атрибуты строки (setLineDash, getLineDash, lineDashOffset)

• теневые атрибуты (shadowColor, shadowOffsetX, shadowOffsetY, shadowBlur)

• прямоугольные методы (clearRect, fillRect, strokeRect)

• методы сложных фигур (beginPath, moveTo, closePath, lineTo, quadraticCurveTo, bezierCurveTo, arcTo, дуга, прямоугольник, заливка, обводка, клип, isPointInPath)

• интерфейс TextMetrics (объект CanvasTextMetrics), его свойства (ширина), а также методы и атрибуты текста (font, textAlign, textBaseline, fillText, strokeText, measureText)

• метод изображений (drawImage)

• интерфейс ImageData (объект CanvasImageData), его свойства (данные, высота и ширина) и методы манипулирования пикселями (createImageDataas, getImageData, putImageData)

• IE11 добавляет свойства (msFillRule, msImageSmoothingEnabled)

• объект CanvasPixelArray