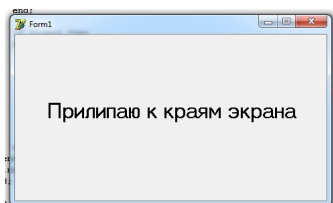


Существует функция API Windows, которая позволяет привязывать окна к краю экрана. Здесь применяются сообщения Windows. Delphi делает обработку сообщений легким, благодаря использованию событий, которые обычно генерируются в ответ на сообщения Windows после того, как она передает его в приложение.



Хотя многие сообщения Windows обрабатываются Delphi, но некоторые необходимо доработать. Например, мы узнаем, если форма меняет размер (при помощи события **OnResize** и обработке сообщения **WM_SIZE**). Но как мы узнаем, что форма была перемещена? Delphi может получить сообщение, но изначально ничего с ним не делает.

Сообщение **WM_MOVING** посылается окну, когда пользователь его перемещает. Обработывая это сообщение, приложение может отслеживать размер и положение перемещаемого окна и при необходимости, изменить его размер или положение.

Сообщение **WM_WINDOWPOSCHANGING** посылается окну, когда меняется его размер, положение или Z-порядок в результате вызова функции **SetWindowPos** или иной функции управления окном.

Простого сообщения нам не достаточно, нам нужно знать и другие параметры. Например, сообщение **WM_MOVE** говорит нам, что изменилась позиция у формы, но оно также имеет параметр **LPARAM**, который может нам дать **X** и **Y** позицию.

У сообщения **WM_WINDOWPOSCHANGING** всего один параметр, который указывает на структуру **WINDOWPOS**, который содержит

информацию о новом размере и положении окна. Структура

WINDOWPOS

определяется так:

```
type TWindowPos = packed record   hwnd: HWND; { идентификатор окна }
hwndInsertAfter: HWND; { окно над ним }   x: Integer; { левый край окна }   y: Integer; {
правый край окна }   cx: Integer; { ширина окна }   cy: Integer; { высота окна }   flags:
UINT; { опции позиции окна } end;
```

В нашем случае, сделаем, чтобы форма прилипла к краю экрана на определенном расстоянии, например, в 20 пикселах от края.

Сообщение WM_WINDOWPOSCHANGING

Поместите объявление процедуры **WMWINDOWPOSCHANGING** в части private формы.

Отменить прилипание к какому-либо краю экрана просто, нужно лишь удалить часть кода, который отвечает за свой участок.

SystemParametersInfo вызывается с **SPI_GETWORKAREA** в качестве первого параметра и получает размер рабочей области экрана (Рабочий стол). Мы будем использовать его для определения полезной площади за исключением Панели Задач, Панелей инструментов и т.д.

Код программы:

```
unit Unit1; interface uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms, Dialogs, StdCtrls; type TForm1 = class(TForm) private {
Private declarations } procedure WMWINDOWPOSCHANGING( var Msg:
TWMWINDOWPOSCHANGING); message WM_WINDOWPOSCHANGING; public {
Public declarations } end; var Form1: TForm1; implementation uses Types; {$R
*.dfm} procedure TForm1.WMWINDOWPOSCHANGING( var Msg:
TWMWINDOWPOSCHANGING); var Docked: boolean; rWorkArea: TRect; StickAt:
Word; // расстояние от края экрана begin Docked:= false; StickAt:= 20; // 20px от края
экрана SystemParametersInfo(SPI_GETWORKAREA, 0, @rWorkArea, 0); with
Msg.WindowPos^ do begin // Прилипание к левому краю if x = rWorkArea.Right -
StickAt then begin x:= rWorkArea.Right - cx; Docked:= true; end; //
Прилипание к верхнему краю if y = rWorkArea.Bottom - StickAt then begin y:=
rWorkArea.Bottom - cy; Docked:= true; end; if Docked then begin with
rWorkArea do begin // Не перемещать за пределы экрана if x > Right then x:=
Right - cx; if y
Bottom then y:= Bottom - cy; end; end; end; inherited; end; end.
```

Теперь запустите проект и перемещайте окно к границам экрана, чтобы посмотреть эффект прилипания.

[Скачать проект](#)