

Есть много ситуаций, когда Вы хотите обменяться информацией между двумя приложениями. Не нужно прибегать к отправке сообщений через TCP и сокет (потому что эти приложения выполняются на одной машине). Вы можете просто послать, и также получить специальное сообщение Windows **WM\_COPYDATA**.

Обработка сообщений Windows в Delphi очень проста: посылая запрос **SendMessage** с заполненными данными

**WM\_COPYDATA**

.

### **WM\_CopyData и TCopyDataStruct**

Сообщение **WM\_COPYDATA** дает Вам возможность послать данные из одного приложения в другое. Приложение получает данные в структуре

**TCopyDataStruct**

.

**TCopyDataStruct**

определен в модуле

*Windows.pas*

и включает в себя структуру

**COPYDATASTRUCT**

, которая содержит данные для передачи.

Вот объявление и описание **TCopyDataStruct**:

```
type TCopyDataStruct = packed record   dwData: DWORD; // до 32 бит, которые
нужно передать           // приложению-получателю   cbData: DWORD; // размер, в
байтах данных, указателя lpData   lpData: Pointer; // Указатель на данные, которые
нужно передать           // приложению-получателю. Может быть NIL.   end;   Пе
```

### **решылка String через WM\_COPYDATA**

Для приложения *Sender*, чтобы послать данные к *Receiver*, **CopyDataStruct** должен быть заполнен и передан, используя функцию

**SendMessage**

. Вот как можно послать строковое значение через

**WM\_COPYDATA**

:

```
procedure TSenderMainForm.SendString(); var stringToSend : string; copyDataStruct : TCopyDataStruct; begin stringToSend := 'About Delphi Programming'; copyDataStruct.dwData := 0; // используем это, чтобы // идентифицировать содержание сообщения copyDataStruct.cbData := 1 + Length(stringToSend); copyDataStruct.lpData := PChar(stringToSend); SendData(copyDataStruct); end;
```

Функция **SendData** находит получателя, используя вызов функции API **FindWindow**:

```
procedure TSenderMainForm.SendData(const copyDataStruct: TCopyDataStruct); var receiverHandle : THandle; res : integer; begin receiverHandle := FindWindow(PChar('TReceiverMainForm'), PChar('ReceiverMainForm')); if receiverHandle = 0 then begin ShowMessage('CopyData Receiver NOT found!'); Exit; end; res := SendMessage(receiverHandle, WM_COPYDATA, Integer(Handle), Integer(@copyDataStruct)); end;
```

В коде выше, приложение *Receiver* было найдено, используя функцию API **FindWindow**, передавая имя класса главной формы.

Обратите внимание: **SendMessage** возвращает целочисленное значение, назначенное кодом, который обработал сообщение **WM\_COPYDATA**  
**A** .

Приложение *Receiver* обрабатывает сообщение **WM\_COPYDATA** как:

```
type TReceiverMainForm = class(TForm) private procedure WMCopyData(var Msg : TWMCopyData); message WM_COPYDATA; { ... } implementation { ... } procedure TReceiverMainForm.WMCopyData(var Msg: TWMCopyData); var s : string; begin s := PChar(Msg.CopyDataStruct.lpData); //Отошлем что-нибудь назад msg.Result := 2006; end;
```

Запись **TWMCopyData** объявляется как:

```
TWMCopyData = packed record Msg: Cardinal; From: HWND; // Дескриптор окна, которое передает данные CopyDataStruct: PCopyDataStruct; // передаваемые данные Result: Longint; // Используйте , чтобы отослать значение назад "Sender" end; Пересылка String, Record или Image
```

Исходный код демонстрирует, как послать строку, запись и даже изображение другому приложению.

Вот пример, как переслать графику **TBitmap**:

```
procedure TSenderMainForm.SendImage(); var ms : TMemoryStream; bmp :
TBitmap; copyDataStruct : TCopyDataStruct; begin ms := TMemoryStream.Create; try
bmp := self.GetFormImage; try bmp.SaveToStream(ms); finally bmp.Free;
end; copyDataStruct.dwData := Integer(cdtImage); // идентифицировать данные
copyDataStruct.cbData := ms.Size; copyDataStruct.lpData := ms.Memory;
SendData(copyDataStruct); finally ms.Free; end; end;
```

Теперь, как получить его:

```
procedure TReceiverMainForm.HandleCopyDataImage( copyDataStruct:
PCopyDataStruct); var ms: TMemoryStream; begin ms := TMemoryStream.Create; try
ms.Write(copyDataStruct.lpData^, copyDataStruct.cbData); ms.Position := 0;
receivedImage.Picture.Bitmap.LoadFromStream(ms); finally ms.Free; end; end;
```

Вот и все!

Перевод с сайта [www.delphi.about.com](http://www.delphi.about.com)