

Как получить IP-адрес компьютера, средствами Delphi через Socket API?

Доброго времени суток, уважаемые читатели блога SoftMaker.kz. Как вам известно, сегодня многие люди хотят иметь доступ к компьютерам локальной сети (LAN) или сети Интернет. Так же многие хотят уметь создавать программы для работы в локальной сети (LAN) или сети Интернет. В статье из этой же серии уже было написано, [как программно получить внешний IP адрес средствами Delphi](#). Сегодня мы рассмотрим ещё одну из наиболее интересных задач это **получение IP-адреса компьютера, подключенному к локальной сети (LAN)**.

Весь интернет построен на TCP/IP связи. Часть TCP описывает, как два компьютера могут установить соединение друг с другом для передачи данных. IP часть отвечает за передачу и маршрутизацию сообщений между узлами Internet. Каждый компьютер, имеет свой уникальный IP-адрес, что позволяет другим узнать путь к любому компьютеру в сети.

Чтобы **получить IP-адрес компьютера**, который вы используете при подключении к локальной сети (LAN) мы должны вызывать некоторые функции API определенные в модуле Winsock. Мы создадим функцию GetIPFromHost, которая вызывает несколько Winsock API функций для того, чтобы **получить IP-адрес**.

Прежде чем использовать WinSock функцию, мы должны открыть сессию. Сессия открывается с помощью функции WSASStartup. В конце функции GetIPFromHost вызывается функции SACKleanup, чтобы прекратить использование сокетов API Windows.

Чтобы **получить IP адрес компьютера, подключённого к локальной сети (LAN)** мы должны использовать функцию GetHostByName в сочетании с функцией GetHostName. Каждый компьютер является

хостом (Host), и мы можем получить имя компьютера воспользовавшись функцией GetHostName. Теперь, получив имя компьютера, мы **получим и его IP-адрес** с помощью функции GetHostByName. Откройте Delphi и бросьте на форму одну кнопку и два поля Edit. Приведенный ниже код нужен, чтобы **получить адрес IP, который использует Ваш компьютер.**

```
uses Winsock;

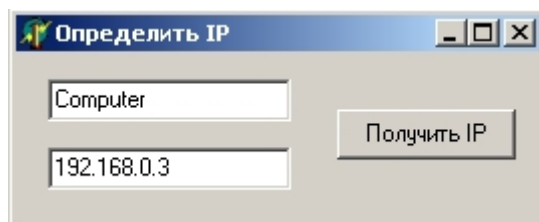
function GetIPFromHost
(var HostName, IPAddr, WSAErr: string): Boolean;
type
  Name = array[0..100] of Char;
  PName = ^Name;
var
  HEnt: pHostEnt;
  HName: PName;
  WSAData: TWSAData;
  i: Integer;
begin
  Result := False;
  if WSASStartup($0101, WSAData) <> 0 then begin
    WSAErr := 'Сокет не отвечает!";
    Exit;
  end;
  IPAddr := '';
  New(HName);

  if GetHostName(HName^, SizeOf(Name)) = 0 then
  begin
    HostName := StrPas(HName^);
    HEnt := GetHostByName(HName^);
    for i := 0 to HEnt^.h_length - 1 do
      IPAddr :=
        Concat(IPAddr,
          IntToStr(Ord(HEnt^.h_addr_list^[i])) + '.');
    SetLength(IPAddr, Length(IPAddr) - 1);
    Result := True;
  end
  else begin
    case WSAGetLastError of
```

```
WSANOTINITIALISED:WSAErr:='WSANotInitialised';
WSAENETDOWN      :WSAErr:='WSAENetDown';
WSAEINPROGRESS   :WSAErr:='WSAEInProgress';
end;
end;
Dispose(HName);
WSACleanup;
end;
```

Пропишите в обработчике события кнопки следующее:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Host, IP, Err: string;
begin
  if GetIPFromHost(Host, IP, Err) then begin
    Edit1.Text := Host;
    Edit2.Text := IP;
  end
  else
    MessageDlg(Err, mtError, [mbOk], 0);
end;
```



Примерно такой же появляется диалог формы, когда выполняется [GetExternalIP – программа для получения внешнего IP адреса](#).

Этот блог читают уже много людей
- читай и ТЫ!

Да, Я тоже хочу читать статьи!

Как программно получить внешний IP адрес средствами Delphi?

Доброго времени суток, уважаемые читатели нашего сайта! Сегодня я приведу пример на Delphi, который поможет получить ваш внешний IP адрес в сети Интернет. В статье из этой же серии уже было написано [как получить IP адрес компьютера в локальной сети с помощью Delphi](#). А здесь приведён пример на Delphi, с помощью которого можно получить IP адрес компьютера даже, если он работает в локальной сети и стоит за прокси сервером, шлюзом или маршрутизатором. Этот функционал реализован в небольшой программке [GetExternalIP](#).

Протокол Internet адреса (Internet Protocol Address) позволяет компьютерам (принтерам, модема) взаимодействовать друг с другом в компьютерной сети.

Весь интернет построен на TCP/IP связи. Часть TCP описывает, как два компьютера могут установить соединение друг с другом для передачи данных. IP часть отвечает за передачу и маршрутизацию сообщений между узлами Internet.

Если Ваш компьютер является частью домашней, либо офисной сети, у Вас, скорее всего, есть какое-то оборудование для связи с Интернет, соединяющее домашнюю (офисную) локальную сеть (LAN) с глобальной сетью (WAN).

Как правило, внутренние (локальные) IP адреса это числа типа: «192.68.1.2». Внешние, (глобальные) IP-адрес модема (например, [роутер KEENETIC](#)), устанавливаются, когда модем подключается к интернет-провайдеру. Если Вы хотите узнать внешний IP-адрес модема, используя Delphi код, можно

воспользоваться услугами сайта «<http://www.softmaker.kz>».

Сначала на форму нужно бросить компонент **IdFTP** из набора компонент **Indy Clients**. Приведенный ниже код нужен, чтобы получить адрес IP модема, который использует Ваш компьютер для подключения к сети Интернет.

```
function GetHTML(const AURL: string): string;
var
  sHead,sHost,sPage: string;
  x,xCnt,xCntTotal: integer;
  sock: TClientSocket;
  ws: TWinSocketStream;
  ss: TStringStream;
  buff: array[0..4095] of char;
const
  CrLf = #13#10;
begin
  Result := '';

  sHost := AURL;
  x := Pos('//',sHost) ;
  if x > 0 then
    System.Delete(sHost,1,x+1) ;
  x := Pos('/',sHost) ;
  if x > 0 then
    begin
      sPage := Copy(sHost,x,Length(sHost)) ;
      System.Delete(sHost,x,Length(sHost)) ;
    end
  else
    begin
      sPage := '/';
    end;

  sock := TClientSocket.Create(nil) ;
  try
    try
      sock.ClientType := ctBlocking;
      sock.Port := 80;
      sock.Host := sHost;
```

```

sock.Open;

// установим задержку на 20 секунд
ws := TWinSocketStream.Create(sock.Socket,20000) ;
ss := TStringStream.Create('') ;
try
    sHead := 'GET ' + sPage + ' HTTP/1.0 ' + CrLf +
            'Host: ' + sHost + CrLf +
CrLf;
    StrPCopy(buff,sHead) ;

    ws.Write(buff,Length(sHead) + 1) ;
    ws.Position := 0;

    FillChar(buff,SizeOf(buff),0) ;
    repeat
        xCnt := ws.Read(buff,SizeOf(buff)) ;
        xCntTotal := xCntTotal + xCnt;
        ss.Write(buff[0],xCnt) ;
    until xCnt = 0;

    Result := ss.DataString;
finally
    ws.Free;
    ss.Free;
end;
except
end;
finally
    sock.Free;
end;
end;

```

Вот как можно использовать эту функцию, чтобы получить внешний IP адрес с сайта www.softmaker.kz.

Бросим на форму кнопку и пропишем в обработчике события следующее:

```

procedure TForm1.Button1Click(Sender: TObject);
var

```

```
    ip: string;
begin
    with TStringlist.Create do
    try
                                                Text :=
GetHTML('http://www.softmaker.kz/get_an_external_ip_address.ph
p') ;
        if Count > 0 then ip := Strings[Count - 1];
    finally
        Free;
    end;
    ShowMessage('Ваш внешний адрес ip: ' + ip);
end;
```

Содержимое файла **get_an_external_ip_address.php** такое:

```
<?php

$ip=$_SERVER['REMOTE_ADDR'];
echo $ip;

?>
```

Этот блог читают уже много людей
- читай и ТЫ!

Да, Я тоже хочу читать статьи!

**Как отправить письмо,
используя Delphi с помощью**

библиотеки Synapse?

Здравствуйте, дорогие посетители! Сегодня поговорим про отправку сообщений средствами Delphi. До этого была написана заметка о том, [как с помощью Delphi получить значения всех элементов Edit диалога формы](#). В двух статьях было описано, как получить с помощью Delphi IP-адрес [тут](#) и [тут](#). А теперь перейдем к нашему вопросу.

Начнём с того, что **скачаем библиотеку Synapse** отсюда: synapse.agarat.cz/doku.php. Распакуем библиотеку во временный каталог. Создадим каталог D:\Delphi\Synapse\. Будет лучше, если названия каталогов содержат латинские символы и без пробелов. Скопируем содержимое каталога source\lib из распакованного архива в каталог D:\Delphi\Synapse\. Теперь установим библиотеку Synapse.

Если у Вас Delphi 5-7, тогда:

- File -> Close All.
- Tools -> Enviroment Options. В появившемся диалоге находим закладку Library.
- Найдём надпись Browsing path. Нажимаем на кнопку с тремя точками [...] справа от него.
- В появившемся диалоге находим кнопку с тремя точками [...]. Нажимаем на эту кнопку, выйдет окно для выбора каталога, находим наш каталог D:\Delphi\Synapse\, выбираем его и нажимаем на кнопку ОК.
- Добавим выбранный каталог, нажав Add и ОК для закрытия окна.
- Теперь для надписи Library Path.
- Найдём надпись Library Path. Нажимаем на кнопку с тремя точками [...] справа от него.
- В появившемся диалоге находим кнопку с тремя точками [...]. Нажимаем на эту кнопку, выйдет окно для выбора каталога, находим наш каталог D:\Delphi\Synapse\,

выбираем его и нажимаем на кнопку ОК.

- Добавим выбранный каталог, нажав Add и ОК для закрытия окна.

Для Delphi 2006-2007 это делается так:

- File -> Close All.
- Tools -> Options. В появившемся диалоге в дереве слева находим Environment options -> Delphi Options -> Library - Win32.
- Найдём надпись Browsing path. Нажимаем на кнопку с тремя точками [...] справа от него.
- В появившемся диалоге находим кнопку с тремя точками [...]. Нажимаем на эту кнопку, выйдет окно для выбора каталога, находим наш каталог D:\Delphi\Synapse\, выбираем его и нажимаем на кнопку ОК.
- Добавим выбранный каталог, нажав Add и ОК для закрытия окна.
- Теперь для надписи Library Path.
- Найдём надпись Library Path. Нажимаем на кнопку с тремя точками [...] справа от него.
- В появившемся диалоге находим кнопку с тремя точками [...]. Нажимаем на эту кнопку, выйдет окно для выбора каталога, находим наш каталог D:\Delphi\Synapse\, выбираем его и нажимаем на кнопку ОК.
- Добавим выбранный каталог, нажав Add и ОК для закрытия окна.

Приступим к **написанию кода по отправке почтового сообщения**. Создадим новый проект. Для того, чтобы отправить письмо необходимо использовать протокол smtp. В **библиотеке Synapse** есть необходимые для этого модули smtpsend, mimemess, mimepart. Подключим эти библиотеки:

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls,  
Forms, Dialogs,
```

```
StdCtrls, httpsend, mimemess, mimepart, smtpsend;
```

Напишем следующую процедуру:

```
Procedure SendMail (Host, Subject, pTo, From , TextBody,
                   HTMLBody, login,password :
string);
var Msg : TMimeMess; // сообщение
    StringList : TStringList; // содержимое письма
    MIMEPart : TMimePart; // части сообщения (на будущее)
begin
    Msg := TMimeMess.Create; // создаем новое сообщение
    StringList := TStringList.Create;
    try
        // Добавляем заголовки
        Msg.Header.Subject := Subject; // тема сообщения
        Msg.Header.From := From; // имя и адрес отправителя
        Msg.Header.ToList.Add(pTo); // имя и адрес получателя
        // создаем корневой элемент
        MIMEPart := Msg.AddPartMultipart('alternative', nil);
        if length(TextBody)=0 then
            // если формат HTML
            begin
                StringList.Text := HTMLBody;
                Msg.AddPartHTML(StringList, MIMEPart);
            end
        else
            // если текстовый формат
            begin
                StringList.Text := TextBody;
                Msg.AddPartText(StringList, MIMEPart);
            end;
            // Кодируем и отправляем
            Msg.EncodeMessage;
            // Отправляем.
            if
smtpsend.SendToRaw(From, pTo, Host, Msg.Lines, login, password)
then
        ShowMessage('Письмо отправлено')
    else
        ShowMessage('Письмо не отправлено');
finally
```

```
Msg.Free;  
StringList.Free;  
end;  
end;
```

Бросим на форму кнопку и пропишем в обработчике события следующее:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    SendMail('smtp.yandex.ru',  
            'Тема тестового письма',  
            ''' + 'komu_pismo' + ''' <' + 'komu@pismo.com' + '>',  
            ''' + 'ot_kogo' + ''' <' + 'otkogo@pismo.ru' + '>',  
            'Содержимое письма в текстовом формате',  
            '',  
            'login_smtp', 'password_smtp')  
end;
```

Этот блог читают уже много людей
- читай и ТЫ!

Да, Я тоже хочу читать статьи!

Как с помощью Delphi получить значения всех элементов Edit диалога формы?

Приветствую Вас, дорогие посетители! На этом блоге уже написана статья, [как на Delphi с помощью библиотеки Synapse отправлять письма](#). Также было рассказано о получении с помощью Delphi IP-адреса [тут](#) и [здесь](#). Эта статья поможет нам

разобраться как прочитать все значения элементов диалога Edit из окна другого приложения, средствами Delphi, используя функции WinAPI. Итого, начнем!

Мы будем пользоваться функциями WinAPI для получения элементов диалога класса Edit. Например, нам нужно получить значения Edit с формы с заголовком «MyProgram», поэтому воспользуемся функцией WinAPI FindWindow.

```
HWND FindWindow(  
    LPCTSTR lpClassName,          // указатель на имя класса  
    LPCTSTR lpWindowName         // указатель на имя окна  
);
```

В первой строке мы запишем:

```
Hndl:=FindWindow(nil, 'MyProgram');
```

Как видим, с помощью этой функции мы получаем указатель на окно (Handle – дескриптор окна). Заносим дескриптор окна в переменную Hndl. Этот Handle нужен нам для следующего действия:

```
Hedt:=FindWindowEx(Hndl,0,'Edit',''); // получаю первый Edit
```

Для поиска воспользуемся функцией WinAPI FindWindowEx (ПоискОкнаУлучшенная):

[Не поддерживается в Windows NT]

Функция FindWindowEx отыскивает данные о дескрипторе окна, имя класса и имя окна которого соответствуют определенным строкам.

Функция поиска дочерних окон начинается с первого до последнего заданного дочернего окна.

```
HWND FindWindowEx  
(
```

```

    HWND hwndParent,           // дескриптор родительского
окна
    HWND hwndChildAfter,      // дескриптор дочернего окна
    LPCTSTR lpszClass,        // указатель имени класса
    LPCTSTR lpszWindow        // указатель имени окна
);

```

Теперь мы получаем в переменную Hedt, дескриптор дочернего окна, первого Edit. Осталось открыть перебор по элементам диалога окна. Для этого используем функцию WinAPI **GetWindow**:

Функция GetWindow возвращает дескриптор окна, которое определено отношением (Z индекса или родительским) к указанному окну.

```

HWND GetWindow
(
    HWND hWnd, // дескриптор родительского окна
    UINT uCmd  // флаг отношений
);

```

Поэтому запишем следующий текст:

```
Hedt:=GetWindow(Hedt,GW_HWNDFIRST);
```

Теперь в переменной Hedt установлен указатель на первый Edit. Параметр GW_HWNDFIRST указывает на выбор первого элемента. Выборка началась, чтобы её продолжить запишем следующее:

```

repeat
    //получаю next Edit
    Hedt:=GetWindow(Hedt,GW_HWNDNEXT);
    if Hedt<>0 then
SendMessage(Hedt,WM_GETTEXT,WPARAM(512),lparam(HoldString));
    HoldString__ := HoldString__ + ' ' + String(HoldString);
until Hedt=0;

```

Теперь в переменной Hedt установлен указатель на следующий Edit. Параметр GW_HWNDNEXT указывает на выбор следующего

элемента. Выборка будет продолжаться, пока не будут перебраны все элементы. Также мы воспользовались функцией WinAPI SendMessage, с помощью которой мы считываем значение элемента диалога.

LRESULT SendMessage(

```
HWND hWnd,           // дескриптор окна назначения  
UINT Msg,           // сообщение  
WPARAM wParam,     // первый параметр сообщения  
LPARAM lParam      // второй параметр сообщения  
);
```

Параметр WM_GETTEXT указывает окну с дескриптором Hedt, что нам нужно считать его значение в переменную HoldString, размером не больше, чем 512. Давайте проверим нашу работу. Создадим новый проект. Бросим кнопку на форму. Вот весь листинг:

```
procedure Form1.Button1Click(Sender: TObject);  
var  
    HoldString: PChar;  
    HoldString__: String;  
    Hndl, Hedt: HWND;  
begin  
    Hndl:=FindWindow(nil, 'MyProgram');  
    Hedt:=FindWindowEx(Hndl,0,'Edit',''); // получаю первый  
Edit  
    Hedt:=GetWindow(Hedt,GW_HWNDFIRST); //указатель на первый  
Edit  
    repeat  
        //устанавливаем указатель на следующий Edit  
        Hedt:=GetWindow(Hedt,GW_HWNDNEXT);  
        if Hedt<>0 then  
SendMessage(Hedt,WM_GETTEXT,WPARAM(512),lparam(HoldString));  
        HoldString__ := HoldString__ + ' ' +  
String(HoldString);  
    until Hedt=0;  
end;
```

Этот блог читают уже много людей
- читай и ТЫ!

Да, Я тоже хочу читать статьи!