

用Delphi实现远程屏幕抓取计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/641/2021\\_2022\\_\\_E7\\_94\\_A8Delphi\\_E5\\_c97\\_641456.htm](https://www.100test.com/kao_ti2020/641/2021_2022__E7_94_A8Delphi_E5_c97_641456.htm)

用Delphi实现远程屏幕抓取 ---- 在网络管理中，有时需要通过监视远程计算机屏幕来了解网上微机的使用情况。虽然，市面上有很多软件可以实现该功能，有些甚至可以进行远程控制，但在使用上缺乏灵活性，如无法指定远程计算机屏幕区域的大小和位置，进而无法在一屏上同时监视多个屏幕。其实，可以用Delphi自行编制一个灵活的远程屏幕抓取工具，简述如下。 ---- 一、软硬件要求。 ---- Windows95/98对等网，用来监视的计算机（以下简称主控机）和被监视的计算机（以下简称受控机）都必须装有TCP/IP协议，并正确配置。如没有网络，也可以在一台计算机上进行调试。 ---- 二、实现方法。 ---- 编制两个应用程序，一个为VClient.exe，装在受控机上，另一个为VServer.exe，装在主控机上。VServer.exe指定要监视的受控机的IP地址和将要在受控机屏幕上抓取区域的大小和位置，并发出屏幕抓取指令给VClient.exe，VClient.exe得到指令后，在受控机屏幕上选取指定区域，生成数据流，将其发回主控机，并在主控机上显示出抓取区域的BMP图象。由以上过程可以看出，该方法的关键有二：一是如何在受控机上进行屏幕抓取，二是如何通过TCP/IP协议在两台计算机中传输数据。 ----

UDP(User Datagram Protocol，意为用户报文协议)是Internet上广泛采用的通信协议之一。与TCP协议不同，它是一种非连接的传输协议，没有确认机制，可靠性不如TCP，但它的效率却比TCP高，用于远程屏幕监视还是比较适合的。同时

，UDP控件不区分服务器端和客户端，只区分发送端和接收端，编程上较为简单，故选用UDP协议，使用Delphi 4.0提供的TNMUDP控件。 ---- 三、创建演示程序。 ---- 第一步，编制VClient.exe文件。新建Delphi工程，将默认窗体的Name属性设为“Client”。加入TNMUDP控件，Name属性设为“CUDP”；LocalPort属性设为“1111”，让控件CUDP监视受控机的1111端口，当有数据发送到该口时，触发控件CUDP的OnDataReceived事件；RemotePort属性设为“2222”，当控件CUDP发送数据时，将数据发到主控机的2222口。 ---- 在implementation后面加入变量定义 const BufSize=2048.{ 发送每一笔数据的缓冲区大小 } var BmpStream:TMemoryStream. LeftSize:Longint.{ 发送每一笔数据后剩余的字节数 } 为Client的OnCreate事件添加代码： procedure TClient.FormCreate(Sender: TObject). begin BmpStream:=TMemoryStream.Create. end. 为Client的OnDestroy事件添加代码： procedure TClient.FormDestroy(Sender: TObject). begin BmpStream.Free. end. 为控件CUDP的OnDataReceived事件添加代码： procedure TClient.CUDPDataReceived(Sender: TComponent. NumberBytes: Integer. FromIP: String). var CtrlCode:array[0..29] of char. Buf:array[0..BufSize-1] of char. TmpStr:string. SendSize,LeftPos,TopPos,RightPos,BottomPos:integer. begin CUDP.ReadBuffer(CtrlCode,NumberBytes){ 读取控制码 } if CtrlCode[0] CtrlCode[1] CtrlCode[2] CtrlCode[3]=\ ' show\ ' then begin { 控制码前4位为“show”表示主控机发出了抓屏指令 } if BmpStream.Size=0 then { 没有数据可发，必须截屏生成数

```

据 } begin TmpStr:=StrPas(CtrlCode).
TmpStr:=Copy(TmpStr,5,Length(TmpStr)-4).
LeftPos:=StrToInt(Copy(TmpStr,1,Pos(\ ' :\ ' ,TmpStr)-1)).
TmpStr:=Copy(TmpStr,Pos(\ ' :\ ' ,TmpStr) 1,Length(TmpStr)
-Pos(\ ' :\ ' ,TmpStr)). TopPos:=StrToInt(Copy(TmpStr,1,Pos(\
' :\ ' ,TmpStr)-1)). TmpStr:=Copy(TmpStr,Pos(\ ' :\ ' ,TmpStr)
1,Length(TmpStr)- Pos(\ ' :\ ' ,TmpStr)).
RightPos:=StrToInt(Copy(TmpStr,1,Pos(\ ' :\ ' ,TmpStr)-1)).
BottomPos:=StrToInt(Copy(TmpStr,Pos(\ ' :\ ' ,TmpStr )
1,Length(TmpStr)-Pos(\ ' :\ ' ,TmpStr))).
ScreenCap(LeftPos,TopPos,RightPos,BottomPos). { 截取屏幕 }
end. if LeftSize>BufSize then SendSize:=BufSize else
SendSize:=LeftSize. BmpStream.ReadBuffer(Buf,SendSize).
LeftSize:=LeftSize-SendSize. if LeftSize=0 then BmpStream.Clear.{
清空流 } CUDP.RemoteHost:=FromIP. { FromIP为主控机IP地址
} CUDP.SendBuffer(Buf,SendSize). { 将数据发到主控机的2222口
} end. end. 其中ScreenCap是自定义函数，截取屏幕指定区域，
代码如下： procedure TClient.ScreenCap(LeftPos,TopPos,
RightPos,BottomPos:integer). var RectWidth,RectHeight:integer.
SourceDC,DestDC,Bhandle:integer. Bitmap:TBitmap. begin
RectWidth:=RightPos-LeftPos. RectHeight:=BottomPos-TopPos.
SourceDC:=CreateDC(\ ' DISPLAY\ ' ,\ ' ,\ ' ,nil).
DestDC:=CreateCompatibleDC(SourceDC).
Bhandle:=CreateCompatibleBitmap(SourceDC,
RectWidth,RectHeight). SelectObject(DestDC,Bhandle).
BitBlt(DestDC,0,0,RectWidth,RectHeight,SourceDC,

```

```
LeftPos,TopPos,SRCCOPY). Bitmap:=TBitmap.Create.  
Bitmap.Handle:=BHandle. BitMap.SaveToStream(BmpStream).  
BmpStream.Position:=0. LeftSize:=BmpStream.Size. Bitmap.Free.  
DeleteDC(DestDC). ReleaseDC(Bhandle,SourceDC). end. 存为  
“ C:VClientClnUnit.pas ” 和 “ C:VClientVClient.dpr ” ， 并编译  
。 100Test 下载频道开通 ， 各类考试题目直接下载。 详细请访  
问 www.100test.com
```