

Freeware Component Review: The Internet Component Suite

William Rouck



There are lots of great components and application source code listings available on the Web. In this review, you'll find out about a very exciting, completely free set of components that makes Internet access a breeze!

THE Internet Component Suite (ICS) is a collection of Internet components written by François Piette. The author includes full source code for all versions of Delphi and C++ Builder. There are 11 components in the package, and they cover all major Internet protocols, including mail, news, FTP, HTTP, and telnet. Even if you're not programming Internet-only applications, many components in this suite can assist you in troubleshooting your Delphi applications that use IP for remote connections. These include Windows Internet Naming Service (WINS) resolution and specifying TCP/IP named resources for database aliasing. Investigating this free component set can prove worthwhile for any Delphi client application, especially if you're looking for Internet functionality without the high cost of many comparable commercial packages.

Installation of the suite is straightforward. I conducted this review using Delphi 3, which required me to create a new package and add the ICS component files to it. For Delphi 1 and 2 (and C++ Builder), simply add each component using the "Add Component" command until the set is fully installed. The component set is written entirely in native Delphi Object Pascal and requires no OCX support for your client applications.

The building-block component of the set is the TWSocket component. It encapsulates support for most Windows WINSOCK.DLL API functions. It supports both TCP and UDP protocols, and the author provides a link to a primer document on his Web page describing the differences between the two. The author includes a Help file that provides an overview of the set and details a few of the components, one of which is TWSocket. You can set socket ports, listen for new remote connections, and connect your application as a client to a remote server by simply dropping this component on a form. To demonstrate the power of TWSocket, the author includes demonstration programs that create chat programs, DNS lookups, and send messages via the telnet port. For a

demonstration, compile and run both the Client demo (CLIDEMO.DPR) and the Server demo (SRVDEMO.DPR) simultaneously (both are a part of this month's Subscriber Downloads at www.pinpub.com/delphi), and send messages to each program. You'll quickly see the results of the communicating programs and how to create your own Internet messaging code.

Mail functions

Internet mail protocols are fully supported by ICS through the TPop3Client and TSMTPClient components. With the included SMTP and POP3 demonstration programs, you can see exactly how to add such functionality to your programs. To test the ICS mail protocol support, compile and launch the demos, and set the SMTP demo to send a message to your own e-mail account. Then set the POP3 demo to connect to your POP3 server, send your ID and password, and check your account's message status. A memo pane is provided to show you the messaging between your client and your Internet service provider.

FTP

FTP functions are included as well, using the TFtpClient component. The included demonstration program shows you how to connect to a remote FTP server, log in, download directory lists, and other routine FTP functions. You might find this function useful for automating the transfer of report files and local database tables to and from your application and the hosting NT server.

PING Support

The TPing component is an example of ICS functions that can be of use to the average remote-database application. For example, if you're writing programs that must communicate with remote database servers over a wide-area network, the TPing component might be useful for remote diagnostics. You could write a simple PING utility for a remote user with your database server IP addresses hard-coded into it. If your application isn't communicating with the host server, you could have your user select such a function from your application and then give you the results over the phone. Or, you could

have your client program test for a successful PING before launching the rest of your code, and alert the user that communications are temporarily inactive. I created a communications test program using the TPing component. It can be found in the sample Delphi 3 code project accompanying this article, "NETDIAGS.DPR". Note that to compile this program, you must have Internet Component Suite installed. The simple user interface is shown in **Figure 1**.

I included buttons for the IBM Web server and the ABC News Web server to demonstrate how easy you can make this for your users. Coding the PING was easy and required just a few lines of code. **Listing 1**, extracted from NETDIAGS, shows you how to launch a PING function with a hard-coded IP address with only three lines of code and a TPing component dropped on your form.

Listing 1. Using TPing.

```
procedure TDemoPingForm.PingABCServerBitBtnClick(  
  Sender: TObject);  
begin  
  ServerToPing := 'ABCNEWS';  
  IPToPing     := '204.202.137.114';  
  PerformPing (ServerToPing, IPToPing);  
end;
```

The manual IP entry button shown in Figure 1 is easy to implement—simply place the user input for the IP address into the "IPToPing" property of TPing. As is also shown in NETDIAGS, you can place a named address in the "Manual Ping" text field, and the TPing component will conduct a DNS lookup for you. For practice, I recommend that you try writing a user-friendly replacement for the Microsoft command-line PING.EXE utility.

Other ICS protocol support components

The other included components, TFingerCli (Finger

protocol), THttpCli (HTTP Protocol), TNntpCli (newsgroup client), TTnCnx (telnet protocol client), TEmulVT (ANSI terminal emulation), and TTnEmulVT (telnet and ANSI terminal emulation)

complete the set of WINSOCK.DLL API support.

The author provides full, free support for the components. He includes instructions on his Web page for how to join his automated mailing list, where users of his components submit questions and comments. There are links to archives of past mailing list questions and answers. The components are frequently updated.

For full Internet protocol support, this component suite deserves a look before spending money on other options. François Piette has compiled a powerful set of components and provides the full source code to show you how he did it.

The Internet Component Suite can be downloaded from François Piette's OverByte home page at <http://www.rtfm.be/fpiette/indexuk.htm>. ▲



ROUCK.ZIP at www.pinpub.com/delphi

William Rouck is a consultant working with California Institute of Technology's Jet Propulsion Laboratory. His systems experience includes inventory management and client/server management information systems. wrouck@pop.jpl.nasa.gov.

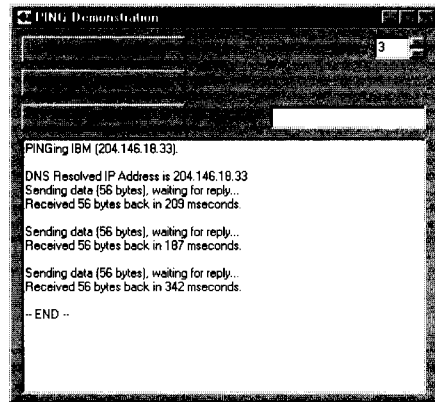


Figure 1. The TPing NETDIAGS demo.