**DatagramSocket**

**The java.net.DatagramSocket class has three constructors:**

public DatagramSocket() throws SocketException

public DatagramSocket(int port) throws SocketException

public DatagramSocket(int port, InetAddress laddr) throws SocketException

The first is used for datagram sockets that are primarily intended to act as clients; that is sockets that will send datagrams before receiving any.

The secned two that specify the port and optionally the IP address of the socket, are primarily intended for servers that must run on a well-known port.

The LocalPortScanner developed earlier only found TCP ports. The following program detects UDP ports in use.

*import java.net.\*;*

*import java.io.IOException;*

*public class NetparamScanner {*

*public static void main(String[] args) {*

*boolean rootaccess = false;*

*for (int port = 1; port < 1024; port += 50) {*

*try {*

*ServerSocket ss = new ServerSocket(port);*

*// if successful*

*rootaccess = true;*

*ss.close();*

*break;*

*}*

*catch (IOException ex) {*

*}*

*}*

*int startport = 1;*

*if (!rootaccess) startport = 1024;*

*int stopport = 65535;*

*for (int port = startport; port <= stopport; port++) {*

*try {*

*DatagramSocket ds = new DatagramSocket(port);*

*ds.close();*

*}*

*catch (IOException ex) {*

*System.out.println("UDP Port " + port + " is occupied.");*

*}*

*}*

*}*

*}*

Since UDP is connectionless it is not possible to write a remote UDP port scanner. The only way you know whether or not a UDP server is listening on a remote port is if it sends something back to you.