

# Auto-discovering Jenkins on the network

If you are developing client applications or server applications that interact with Jenkins, you can benefit from auto-discovering Jenkins in the network. Jenkins supports several modes of programmable auto-discovery.

## UDP multicast/broadcast

Jenkins listens on UDP port 33848. You can either send an UDP broadcast packet (targeted to 255.255.255.255) to this port, or you can send an UDP multicast packet (targeted to 239.77.124.213) to this port. When a packet is received (the payload doesn't matter), Jenkins will send an XML response to the sending UDP port as a datagram. By sending a packet and listening for incoming ones, you can have an UDP client to discover nearby Jenkins instances. UDP multicast works with version 1.335 and later, UDP broadcast works with version 1.282 and later.

The structure of the XML is as follows:

```
<hudson>
  <version>1.354</version><!-- version of the Jenkins -->
  <url>http://server/hudson/</url><!-- the top page of the Jenkins -->
  <slave-port>12345</slave-port><!-- TCP port number for slaves and CLIs to connect to -->
  ... more elements may appear here ...
</hudson>
```

Plugins can extend what to include in this XML fragment by contributing to the [UDPBroadcastFragment](#) extension point.

This feature can be disabled via the [hudson.udp](#) system property.

## From Java

See [the sample code](#) written in Java that does auto-discovery via UDP broadcast.

## DNS multicast

Jenkins advertises itself at "\_jenkins.\_tcp.local" with DNS multicast. This is another way to discover Jenkins instances on the same subnet (or by extending the "\_jenkins.\_tcp" convention, you can discover Jenkins for arbitrary domain names, such as "\_jenkins.\_tcp.example.com", as a convention.) This works with version 1.359 and later.

The same version, url, and slave-port are available as the key/value pairs associated with the DNS record.

This feature can be disabled via the [hudson.DNSMultiCast.disabled](#) system property.

## From Java

You can use [jmdns](#) if you want to perform DNS multi-cast service discovery from Java. The sample code is below:

```
import javax.jmdns.*;

static class SampleListener implements ServiceListener {
    public void addService(JmDNS jmdns, String type, String name) {
        System.out.println("ADD: " + jmdns.getServiceInfo(type, name));
    }
    public void removeService(JmDNS jmdns, String type, String name) {
        System.out.println("REMOVE: " + name);
    }
    public void resolveService(JmDNS jmdns, String type, String name, ServiceInfo info) {
        System.out.println("RESOLVED: " + info);
    }
}

JmDNS jmdns = new JmDNS();
jmdns.addServiceListener("_jenkins._tcp.local.", new SampleListener());
```