Re: InetAddress.equals() Anomoly?

Source: http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.vjsharp/2005−01/0019.html

---

From: Bruno Jouhier [MVP] (*bjouhier_at_club−internet.fr*)
Date: 01/08/05

```
Date: Sat, 8 Jan 2005 09:31:14 +0100
```

I don't agree.

Two objects are equal if they can be used interchangeably and if they produce the same results when any arbitrary method is applied to them.

In your case:
dnsaddr.toString() produces firenet.us/12.34.75.101
ipaddr.toString() produces grande.rivcom.net/12.34.75.101

So, the objects are not equal because they don't produce the same result when you apply toString() to them.

If you consider that two InetAddress should be equal as soon as they have the same IP, you take a biased view, you consider that your equals method should take the IP part into account and completely ignore the name side of the INetAddress object. This is wrong. If you start overriding equals to give it "partially equals" semantics, you are going to run into some trouble.

So, this is not a bug in the JDK. If you want to use the IP as a hash key, you should use the IP (formatted as a string, or as an integer) as key, you should not use InetAddress because this is a richer object that carries more information than an IP.

Bruno.

"Pete Loveall" <psl@ametx.com.NO_SPAM> a écrit dans le message de news: ePrGKCN9EHA.1296@TK2MSFTNGP10.phx.gbl...
> Definite J# bug!
>
> The following output was produced by the code below:
>
> 203574117 203574117
> firenet.us/12.34.75.101=grande.rivcom.net/12.34.75.101−>false
> grande.rivcom.net/12.34.75.101=firenet.us/12.34.75.101−>false
>
> import java.net.;*
>

```
> public class Class1
> {
> public Class1()
> {
> }
> /** @attribute System.STAThread() */
> public static void main(String[] args)
> {
> InetAddress dnsaddr = null;
> InetAddress ipaddr = null;
> try
> {
> dnsaddr = InetAddress.getByName("firenet.us");
> ipaddr = InetAddress.get
```