How does SQLyog connect to MySQL?: How do connection names resolve to ip's?

NOTE: This is not a typical FAQ article.

It does not answer questions about the SQLyog program as such. But we experience quite often that SQLyog user having problems in getting the connection to one ore more MySQL servers working are missing the basic understanding of the TCP protocol that is used for connection. So this is a 'background article' that supplements the more typical FAQ items here.

The TCP protocol needs an ip to connect to a host. An ip is a group of 4 2-digit hexadecimal numbers. Like '0A.1B.2D.3D'. It is common to express this in decimal format 'v.x.y.z' where v,x,y and z are numbers between 0 and 255. To get your own ip in this format just type (on windows) 'ipconfig' on a command-line and the ip is returned.

The ip returned can be a global ip (accessible directly from the internet) or - if you are behind a router - a local ip that only applies behind the router. All computers behind the router share the global ip of the router itself. The router has the functionality to ensure that communication from each machine behind it to and from the internet works.

The TCP-protocol communicates on 'ports'. There are about 65.000 ports possible. A TCP-port is nothing physical but just a number that each 'packet' of information that is sent is coded with. Various ports are reserved for various server programs and programs with sort-of server functionality. The complete list is here:

So to access a MySQL server (running on the standard MySQL port 3306) on the ip v.x.y.z the client must ask for connection to the server 'v.x.y.z:3306'. The Internet can handle this. This is the basically how the Internet and the TCP-protocol works! Pretty simple actually! And there is one basic rule more: the ip 127.0.0.1 is always any computer itself!

However you probably never enter internet addresses that way no matter if it is a website, an FTP-server or a MySQL-server that you access. You use NAMES. However these names must be resolved to the above format. If not resolved to ip's the servers on the Internet can't handle the request.

There are basically four ways of resolving a name to an ip (and in that order):

1) using client hosts' file
2) using (local) nameserver lookup
3) using Domain Name Server (DNS) lookup.
4) using Remote Network routing (using either simple routing based on ports, hosts file or nameserver) on remote network.

1) Client hosts' file
There is an excellent article on the hosts file here. Try to find your own hosts file. You will find this line in it:

How does SQLyog connect to MySQL?: How do connection names resolve to ip's?

127.0.0.1 localhost

You might find more lines too - for several reasons: some viruses and spyware add items to the hosts file. Some users and Sys Admins do it on purpose. But since you are able to connect to a MySQL server running on your local machine with the host name **'localhost'** it is because of the above line in the hosts file. The hosts files resolves the name 'localhost' to ip 127.0.0.1. If you prefer to use **'cutie'** instead of **'localhost'** then just add the line

127.0.0.1 cutie

to your hosts file!

2) A local nameserver.
A local nameserver can be installed explicitly or implicitly. If you have more windows computers connected on a windows' network (using the TCP protocol - not old NETBEUI protocol etc) you already have a local nameserver that was implicitly installed. If you have two machines on your network named 'Bonnie' and 'Clyde', then you can connect to the MySQL server running on 'Bonnie' from the computer 'Clyde' with SQLyog entering 'Bonnie' as the host name. Because the nameserver that was implicitly installed as a part of the Windows network resolves the name 'Bonnie' to 'Bonnie's ip on the network

3) A Domain Name Server
Domain name Servers are part of the Internet structure. Any name/string in the format domain.topleveldomain (ie: school.edu, business.com, whisky.org etc.) can be looked up on the Domain Name Servers of the Internet. The ip returned is then used for creating the connection.

A typical example.

You connect to a remote server with the host name 'mydb.thisismyisp.com'. What happens here? Simply:

First: local hosts file is checked if there are any matching entry. if not proceed to step 2.
Second: local nameserver (if exists) is checked if there are any matching entry. if not proceed to step 3.
Third: Domain Name Server lookup identifies the ip of domain-host 'thisismyisp.com'
Fourth: local network routing systems (router, nameserver or hosts file) on the remote network identifies the local ip mapped to name 'mydb´'

Voila! 'mydb.thisismyisp.com' is resolved to ip v.x.y.z on the internet and ip a.b.c.d on the remote network. Once connection is established the routers and other networking gear in the complete communications chain keeps that information until it is closed or times-out due to inactivity (or some error occur!)

How does SQLyog connect to MySQL?: How do connection names resolve to ip's?

And finally let me remind you of one detail: When connecting to the MySQL server at an ISP it is common practice that MySQL user names and MySQL database names must be prefix'ed with the domain user name (the one that is used for FTP for instance). Like me_username and me_mydb. Simple because other users may have created a MySQL user **username** and a MySQL database **mydb** But failure to do so does not result in a connection error, but an authentication error! Also note that MySQL usernames can be up to 16 characters long only. Read this.

Nevertheless, everything can get 'messed up' if it is the first time you try to connect to a remote MySQL server! And here even tunneling is not involved!

Unique solution ID: #1052
Author: Peter Laursen
Last update: 2005-10-09 12:13