

# MySQL IPv6 enabled

rafael.martinez@novagnet.com  
nGn systems for Euro6IX project-2004  
<http://www.euro6ix.org>

2nd July 2004

## Abstract

In the known area of web database applications, the hat trick composed by “**Linux+Apache +PHP+MySQL**” (also known as formula *LAPM*) is perhaps the most widely used among developers. “**Linux**”, “**Apache**” and “**PHP**” are known already to be available for IPv6. This patch enables MySQL front-end IP subsystem to run on this new protocol.

## Contents

<b>1 Quick install</b>	<b>2</b>
<b>2 Does MySQL stand for dual stack (IPv4-IPv6)?</b>	<b>3</b>
2.1 Client library. . . . .	3
2.2 Daemon server. . . . .	3
<b>3 ChangeLog</b>	<b>4</b>
<b>4 Known Bugs.</b>	<b>4</b>
4.1 MySQL host-dependent privileges transiently bypassed: . . . . .	4
4.2 SSL modes, Test-suite of Perl programs. . . . .	4
4.3 Selective address for binding daemon. . . . .	4
4.4 Others. . . . .	4
<b>5 TODO:</b>	<b>5</b>
5.1 Fixing MySQL privilege system. . . . .	5
<b>6 What to do in a case of fatal failure.</b>	<b>5</b>

**WARNING: THIS IS Beta-Code. NOT RELEASED FOR PRODUCTION PURPOSES, RATHER FOR TESTING. USE AT YOUR OWN RISK. READ THE “KNOWN BUGS” SECTION AND BE WARNED: MySQL IP-dependent PRIVILEGE SYSTEM [1] IS TRANSIENTLY BYPASSED!!!.**

# 1 Quick install

In general terms, systems compiling original MySQL-4.0.8 should support the patched version with no major problems. Please, make sure your software-tools match the corresponding versions :

- `mysql-4.0.8.tar.gz` (<http://dev.mysql.com/downloads/mysql/4.0.html>)
- `gcc, g++` from `gcc-3.2.tar.gz` compiler's suite. (<ftp://ftp.gnu.org>)
- `libtool-1.5.6.tar.gz` (<ftp://ftp.gnu.org>)
- `autoconf-2.57.tar.gz` (<ftp://ftp.gnu.org>)
- `automake-1.7.6.tar.gz` (<ftp://ftp.gnu.org>)
- ...

Our test-system was SuSE-Linux-8.1 (updating libtool, autoconf, automake to the required versions), failing with SuSE-Linux-8.0. However, it is desirable a more exhaustive testing process. Contribute and e-mail the above address!!!

Some skill on installing ordinary MySQL is required. Please, in case of trouble refer the ordinary installation manual[1, 4]. Purpose here is not to teach either MySQL administration or SQL basics.

```
#cd /usr/local/src (*)
#tar xzvf mysql-4.08.tar.gz
#patch -p0 < /path/to/mysql-4.08-v6-20040701.diff
#cd mysql-4.08
#libtoolize --force
#autoreconf
#./configure --enable-ipv6 --prefix=/usr/local/mysql (*)
#make
#make install
#cp support-files/my-medium.cnf /etc/my.cnf
```

Replace (\*) at any one of your choice. To test your new server running on IPv6, proceed to setup the initial-database. Assuming the above paths:

```
#add user mysql
#add group mysql
#/usr/local/sql/bin/mysql_install_db
#cd /usr/local/mysql
#chown -R root .
#chown -R mysql var
#chgrp -R mysql var
```

Be warned of the “root” and “anonymous” accounts created. You can later securing them[3]. Some practical commands to deal with (please, refer to Manual of MySQL to get a more detailed info).

Starting/stopping server:

```
#!/usr/local/mysql/bin/mysqld_safe --user=mysql (the first time)
#!/usr/local/mysql/share/mysql/mysql.server start
#!/usr/local/mysql/share/mysql/mysql.server stop
#!/usr/local/mysql/share/mysql/mysql.server restart
```

Playing some client utilities:

```
>export PATH=/usr/local/mysql/bin:$PATH
>mysqladmin -h 2003:800:4a05:3f3a:427f::3 version
>mysql -h 2003:800:4a05:3f3a:427f::3 -u root -p
mysql> show databases;
mysql> SELECT VERSION(), CURRENT_DATE;
mysql> use mysql;
mysql> show tables;
```

In [5] you can read on how to make the server to start automatically.

## 2 Does MySQL stand for dual stack (IPv4-IPv6)?

As the transition to IPv6 occurs, it is unlikely that IPv4 will be discarded. Instead, an hybrid system running both IPv4-IPv6, would be preferable [6]. Concerning this issue, MySQL IPv6 version is ready for transition under the following circumstances:

### 2.1 Client library.

Released client-libraries are ready to deal with multiple address resolution (DNS AAAA, A registers), i.e, if you try to access a server whose DNS solves both into IPv6 and IPv4 addresses, but the MySQL server runs only on IPv4 or you have no IPv6 connectivity, the client will try first IPv6, and then fall-back into IPv4 in case of failure. Of course, DNS-IPv4 only resolutions are processed as expected.

### 2.2 Daemon server.

For daemon server, things are a bit more complicated. By default, you will bind the given address (IN6\_ADDR\_ANY (::) in most cases, if no other is provided). Being able to process IPv6 connections, this means you will be able to listen IPv4 connections, thanks to “mapped-address” mechanism implemented by the kernel. IPv4 originating addresses will be mapped into IPv6-IPv4 mapped addresses (i.e ::ffff:10.0.0.4). See [6]for a detailed description on this feature.

However, you should not trust this way to get a dual-stack IPv4/IPv6 MySQL server running, since “mapped addresses” are being dropped from some kernel implementations, as OpenBSD family (2004-05-05). You would do better

by providing MySQL an specific IPv4 address to bind on (i.e, IN\_ADDR\_ANY -0.0.0.0- or 10.0.0.34)<sup>1</sup>.

### 3 ChangeLog

- First release: 2004-07-01

### 4 Known Bugs.

#### 4.1 MySQL host-dependent privileges transiently bypassed:

MySQL system implements its own privilege system[2] relying on

- the user trying to connect into the server
- the database the user wants to connect, and
- the IP from which the user tries to

This transient patch **WILL IGNORE** whatever constraints you may setup concerning third item as a MySQL administrator, i.e, effect will be the same as if a wild char (%) was applied. Be warned.

However, the rest of constraints remains full operative, depending on MySQL administrator's policies.

#### 4.2 SSL modes, Test-suite of Perl programs.

To test. Contribute !!

An operative Perl-IPv6 platform is required. For those trying to, get from <http://www.cpan.org> modules INET6 and Socket6.

#### 4.3 Selective address for binding daemon.

Actually it only binds into (IN6\_ADDR\_ANY, ::) address , no matter which address is passed. A patch is coming to fix this bug.

#### 4.4 Others.

Please, report them.

---

<sup>1</sup>See "know bugs" to check the status of development, concerning this point.

## 5 TODO:

Please, send comments, contributions, suggestions to above address or irc news about MySQL...

- Fix the system in order to let it bind addresses other than ANY (::), passed by command-options
- Info and related doc to be updated
- Upload into official CVS repository
- The “configure” script. Apart from “-enable-ipv6”, we should let the user selectively to set “multi-address” resolution (-enable-multi-address, default, safer) or simple one (-disable-multi-address,faster). Actually, it performs first, since I have no skill on M4 language. (It has sense only when enable-ipv6 is selected).

### 5.1 Fixing MySQL privilege system.

Fixing the IP MySQL privilege system is not so a trivial task, beyond the code-porting on network routines and structures. Reasons are explained below:

- On ordinary version, the field of the network address “struct in\_addr.s\_addr” is used as an integer (u\_int32\_t) to index a hash table in a strongly IP-dependent manner. Hence, it is non straightly portable, since we must consider both types of addresses IPv4/IPv6.
- Let’s consider the “mapped addresses” to include IPv4 address as a case inside the IPv6. In such a case, you must consider an integer type of 128 bits!. The ANSI largest integer type is 64 bits (long long), unless we use the arbitrary precise integer library gmp.
- We can consider a double-hash table. The first item is indexed by the first 64 bits (long,long), holding this item a new table indexed by the last 64 bits. Since the hash-table is implemented dynamically, it is spected not to waste  $(2^{129} - 1)$  memory cells.
- Other languages like Java, implement their own HashTables data type. I’m not sure, but GNU C-library seems to implement a hash-table indexed on a string. We could take advantage of this to index the table on the string representation of the IPv6 address (“2003:800:4a05:3f3a:427f::3”)

## 6 What to do in a case of fatal failure.

First at all, be warned again. This is not a production release, but a testing one. However, I did never got a failure. Most part of the times this is due to a poor skill on MySQL administration issues (Neither am I an expert).

But if you are in trouble and you want to recover data, proceed as follows: Just install the ORDINARY MySQL version, as described above, and after installing, DO NOT INITIALIZE the database. This IPv6 patch only modifies the IP front-end, not the internals of the DBMS.

```
#cd /usr/local/src (*)
#tar xzvf mysql-4.08.tar.gz
#cd mysql-4.08
#./configure --prefix=/usr/local/mysql (*)
#make
#make install
```

Then use the client tools and server as ordinary IPv4.

## References

- [1] Manual of contents. <http://www.mysql.org/>
- [2] Manual of contents, 5.4 MySQL Privilege System
- [3] Manual of contents, 5.5.2 Securing the initial MySQL Accounts.
- [4] Manual of contents, 2. Installing MySQL
- [5] Manual of contents, 2.4.2.2 Starting and Stopping MySQL Automatically
- [6] J. Kennedy, An introduction to Writing Protocol Independent Code.  
<http://www.gmonline.demon.co.uk/cscene/CS6/CD6-03.html>