Persistent
Security

# *SafeGuard LM* 5.6 Product Activation Server

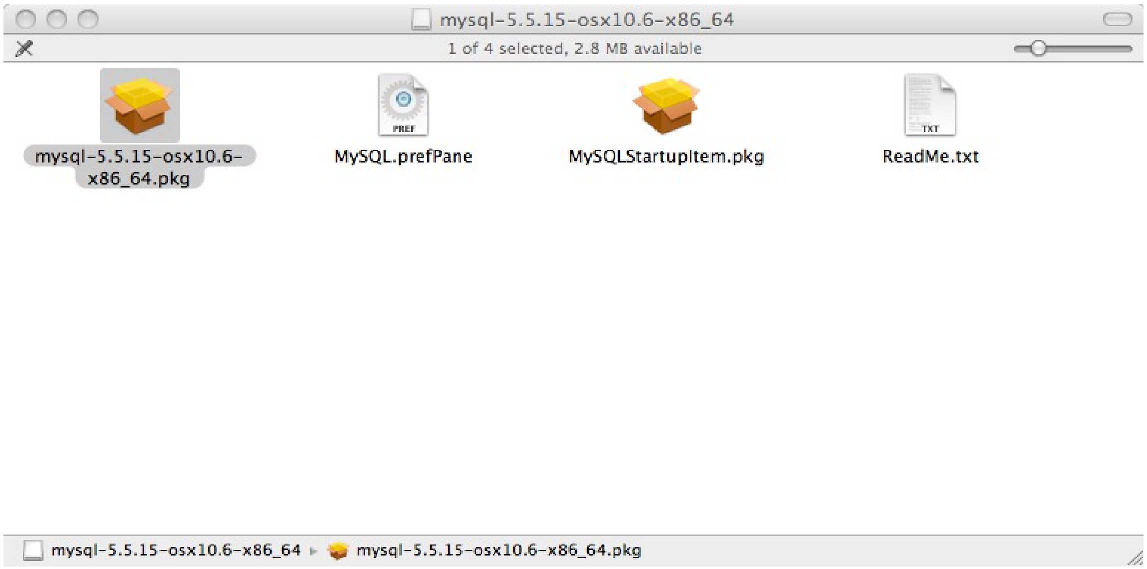
(25-apr-2016)

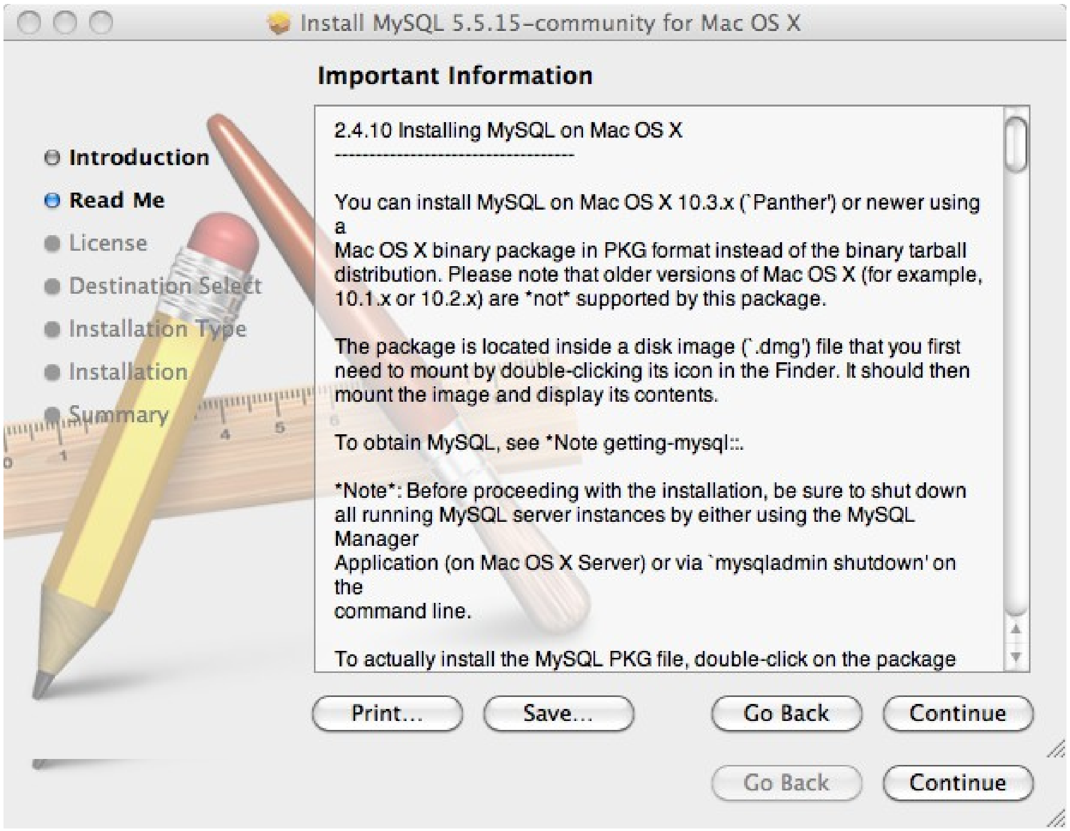
The Product Activation Server uses MySQL to store information that controls license activation. MySQL must be installed for the Product Activation Server to work. The following steps should be taken to install MySQL before running the PAS.

Install MySQL Community Server 5.5 or newer from http://dev.mysql.com/downloads for your platform.

The following will get the Community server and MySQL Workbench installed. Although this example shows the installation for OS X, the procedure is similar for each platform.

Launch the community server installation


mysql-5.5.15-osx10.6-x86_64
1 of 4 selected, 2.8 MB available
mysql-5.5.15-osx10.6-x86_64.pkg
MySQL.prefPane
MySQLStartupItem.pkg
ReadMe.txt
mysql-5.5.15-osx10.6-x86_64 ▸ mysql-5.5.15-osx10.6-x86_64.pkg



Important Information

2.4.10 Installing MySQL on Mac OS X
-----------------------------------

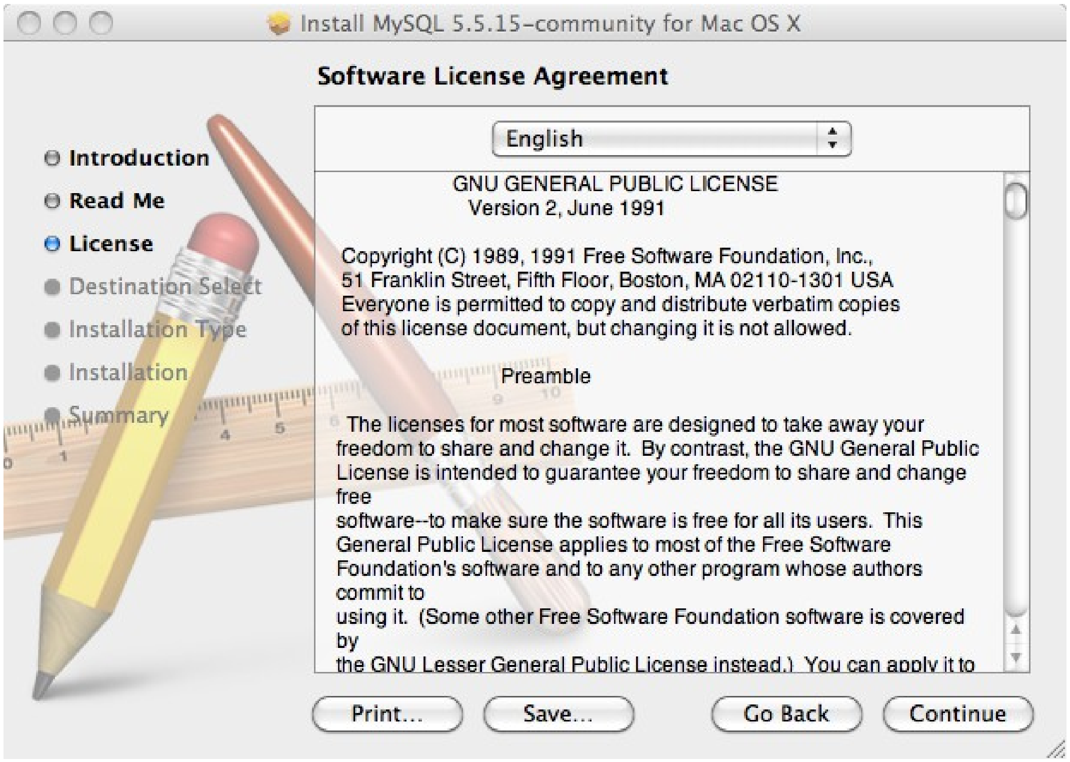
You can install MySQL on Mac OS X 10.3.x (`Panther') or newer using
a
Mac OS X binary package in PKG format instead of the binary tarball
distribution. Please note that older versions of Mac OS X (for example,
10.1.x or 10.2.x) are *not* supported by this package.

The package is located inside a disk image (`.dmg') file that you first
need to mount by double-clicking its icon in the Finder. It should then
mount the image and display its contents.

To obtain MySQL, see *Note getting-mysql::.

*Note*: Before proceeding with the installation, be sure to shut down
all running MySQL server instances by either using the MySQL
Manager
Application (on Mac OS X Server) or via `mysqladmin shutdown' on
the
command line.

To actually install the MySQL PKG file, double-click on the package



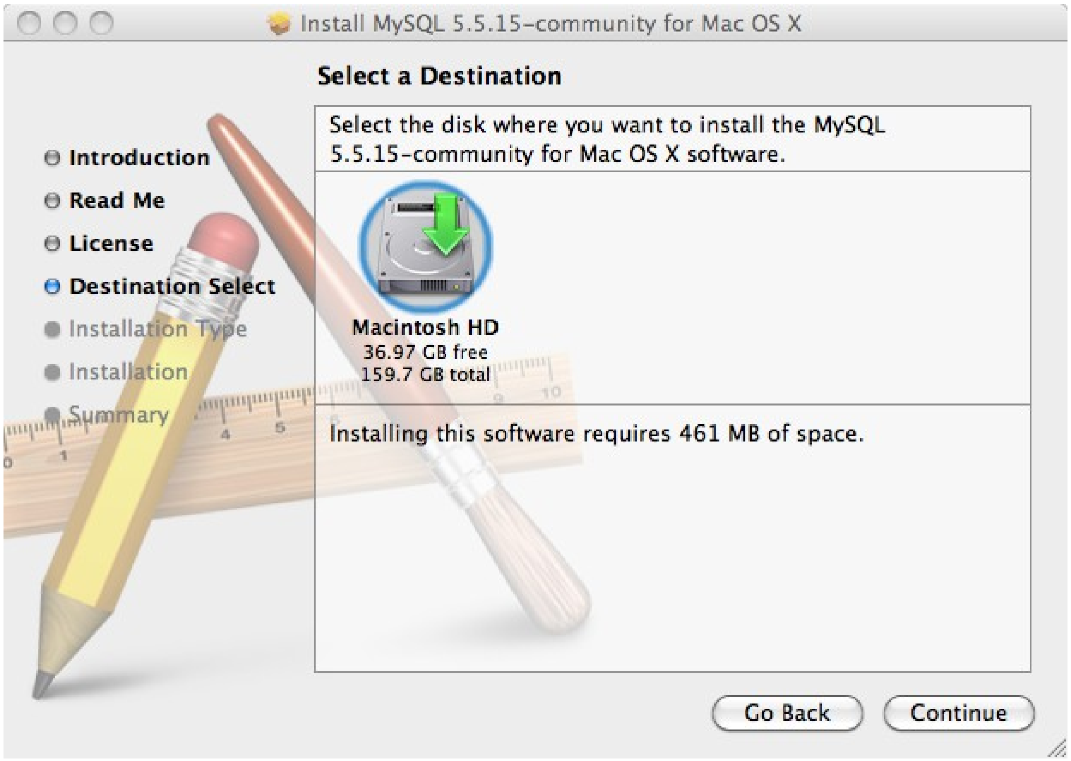
Software License Agreement

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change
free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors
commit to
using it. (Some other Free Software Foundation software is covered
by
the GNU Lesser General Public License instead.) You can apply it to

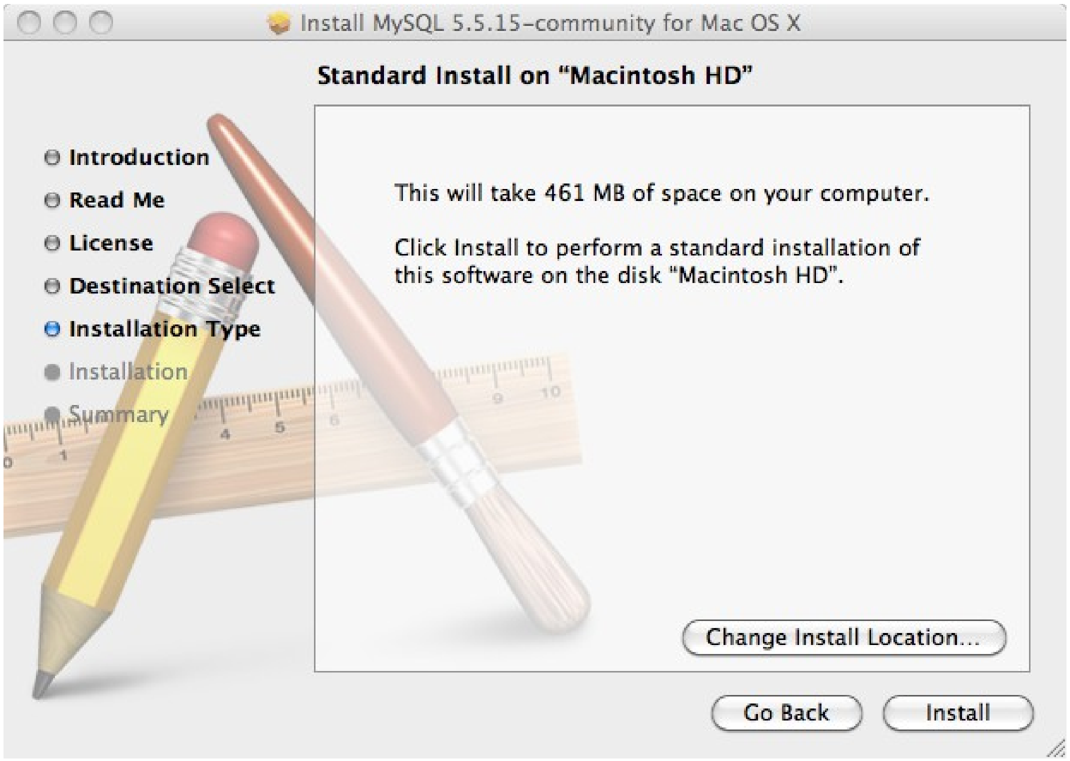


Select a Destination

Select the disk where you want to install the MySQL 5.5.15-community for Mac OS X software.

Macintosh HD
36.97 GB free
159.7 GB total

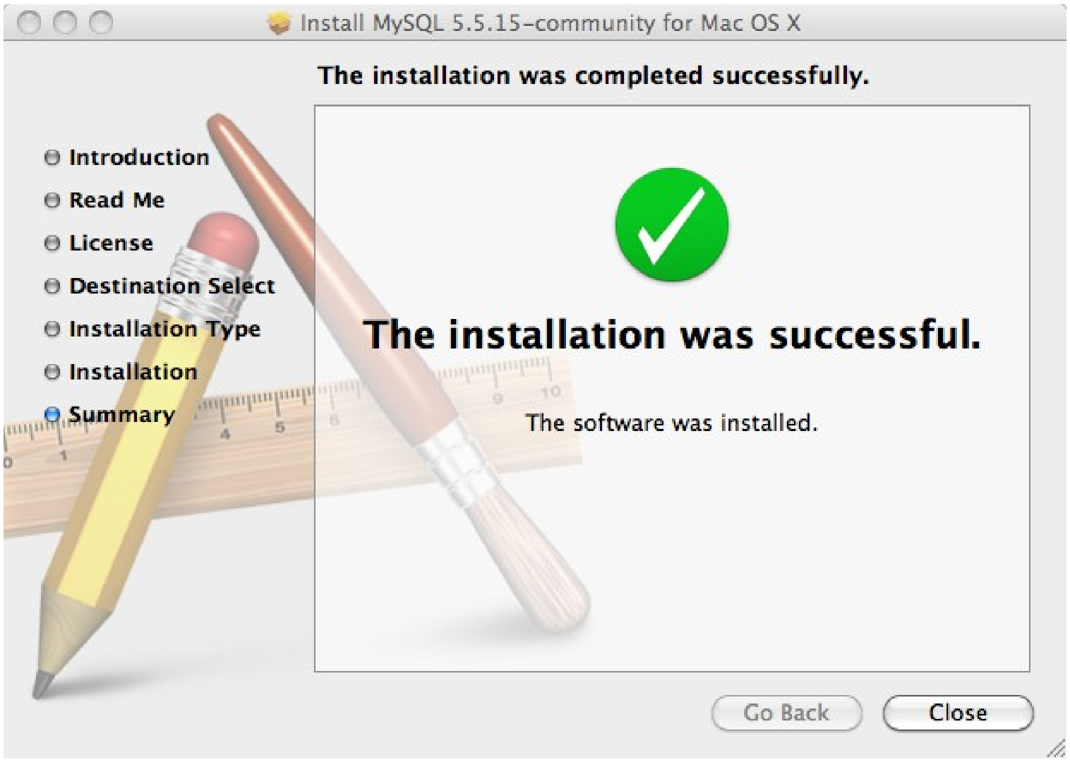
Installing this software requires 461 MB of space.



Standard Install on “Macintosh HD”

This will take 461 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk “Macintosh HD”.



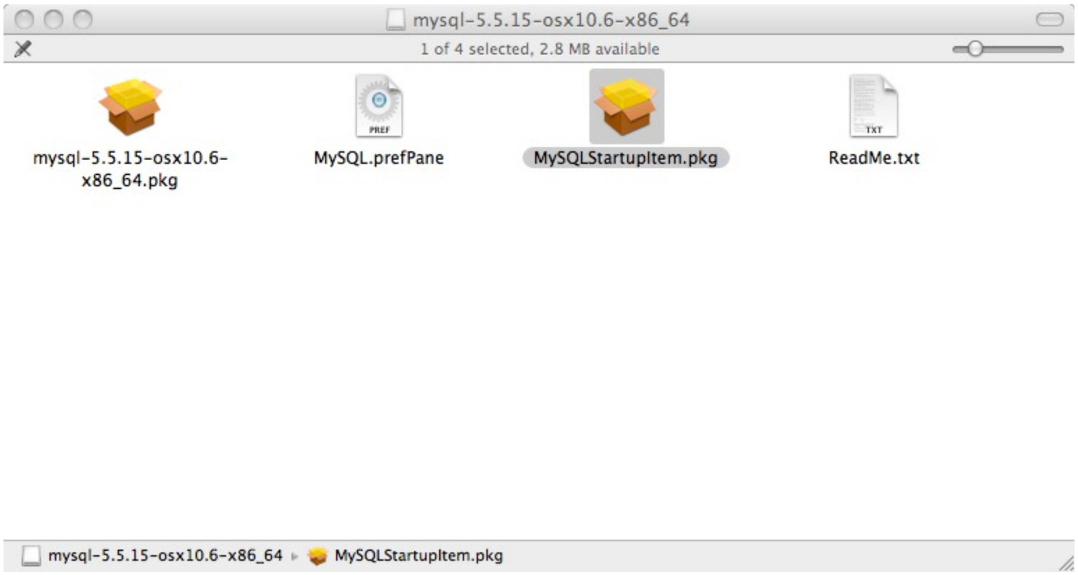
Install MySQL 5.5.15-community for Mac OS X

The installation was completed successfully.

The installation was successful.

The software was installed.

Launch the community server Startup Item installation

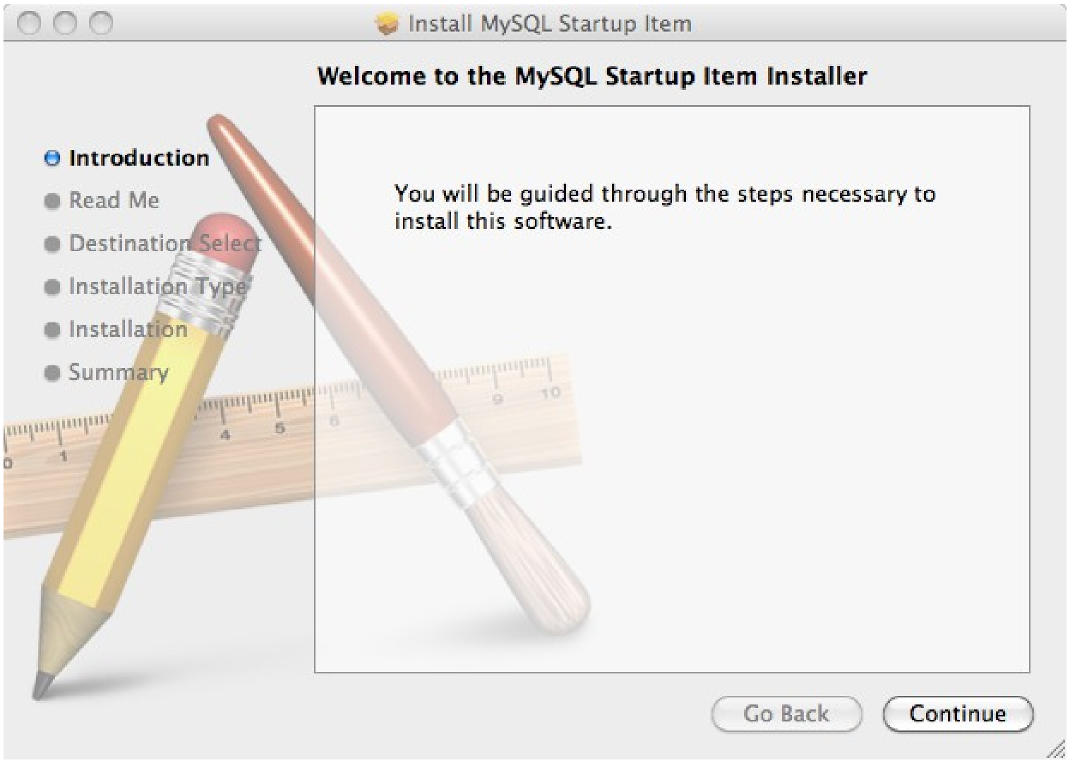


mysql-5.5.15-osx10.6-x86_64

1 of 4 selected, 2.8 MB available

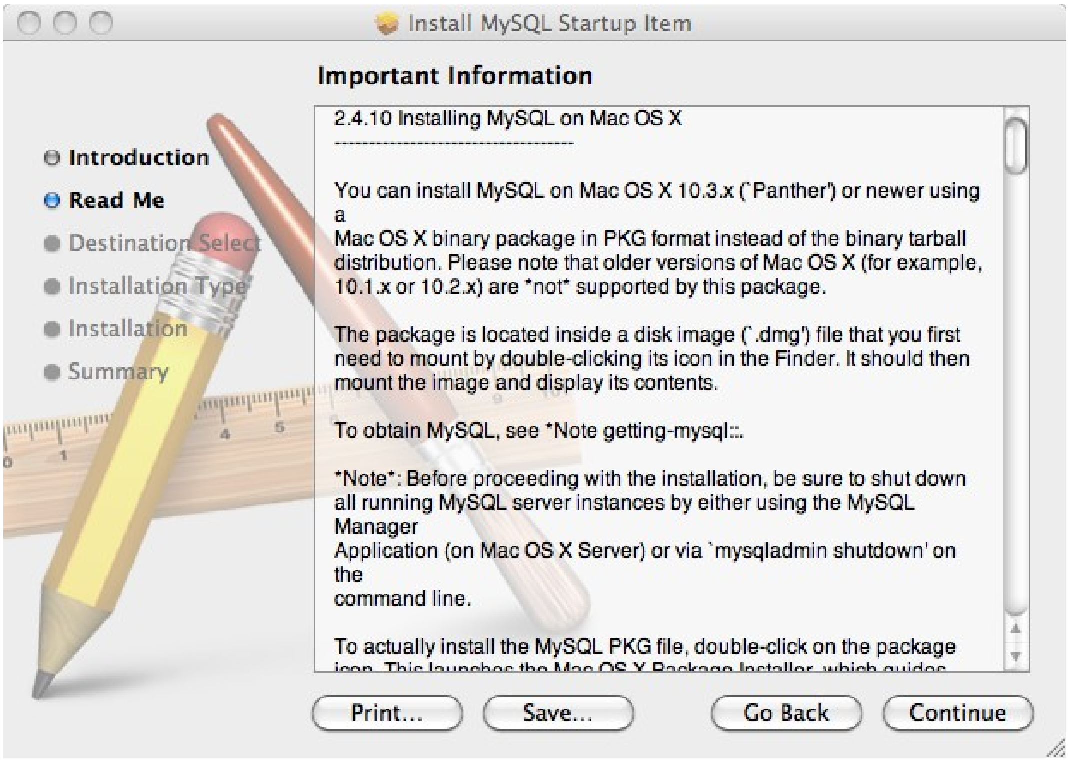
mysql-5.5.15-osx10.6-x86_64.pkg
MySQL.prefPane
MySQLStartupItem.pkg
ReadMe.txt
mysql-5.5.15-osx10.6-x86_64 ▸ MySQLStartupItem.pkg

Take all the defaults



Welcome to the MySQL Startup Item Installer

You will be guided through the steps necessary to install this software.



Important Information

2.4.10 Installing MySQL on Mac OS X
------------------------------------

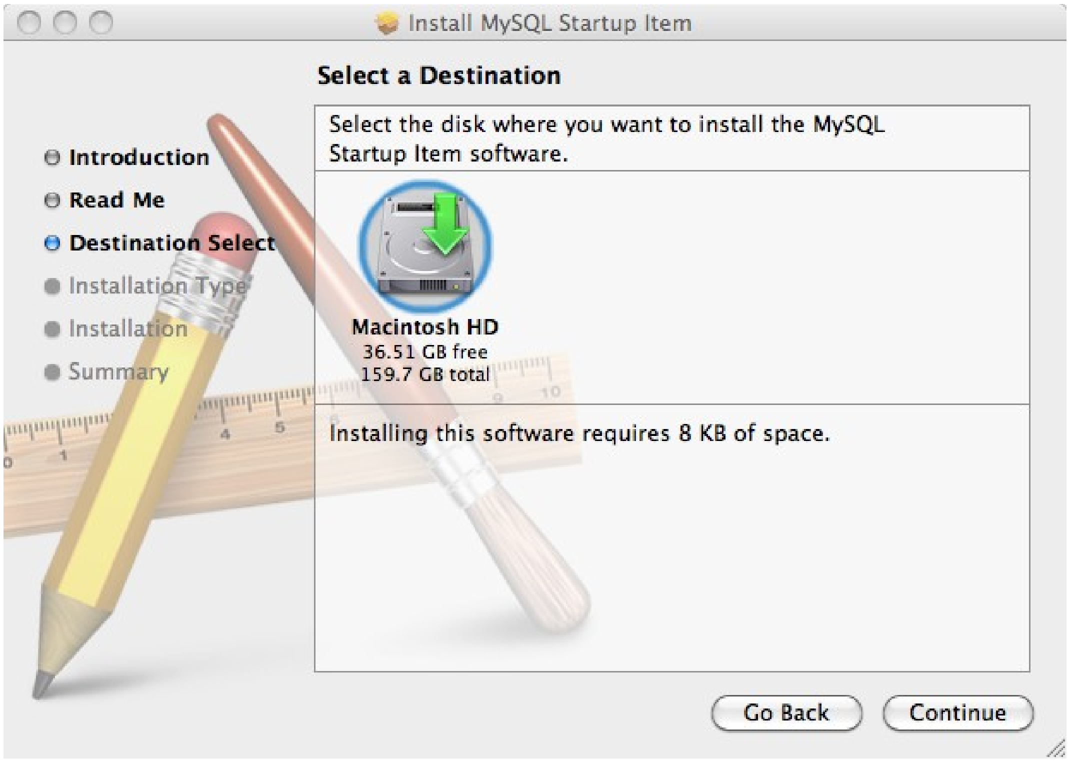
You can install MySQL on Mac OS X 10.3.x (`Panther') or newer using a
Mac OS X binary package in PKG format instead of the binary tarball distribution. Please note that older versions of Mac OS X (for example, 10.1.x or 10.2.x) are *not* supported by this package.

The package is located inside a disk image (`.dmg') file that you first need to mount by double-clicking its icon in the Finder. It should then mount the image and display its contents.

To obtain MySQL, see *Note getting-mysql::.

*Note*: Before proceeding with the installation, be sure to shut down all running MySQL server instances by either using the MySQL Manager
Application (on Mac OS X Server) or via `mysqladmin shutdown' on the
command line.

To actually install the MySQL PKG file, double-click on the package icon. This launches the Mac OS X Package Installer, which guides

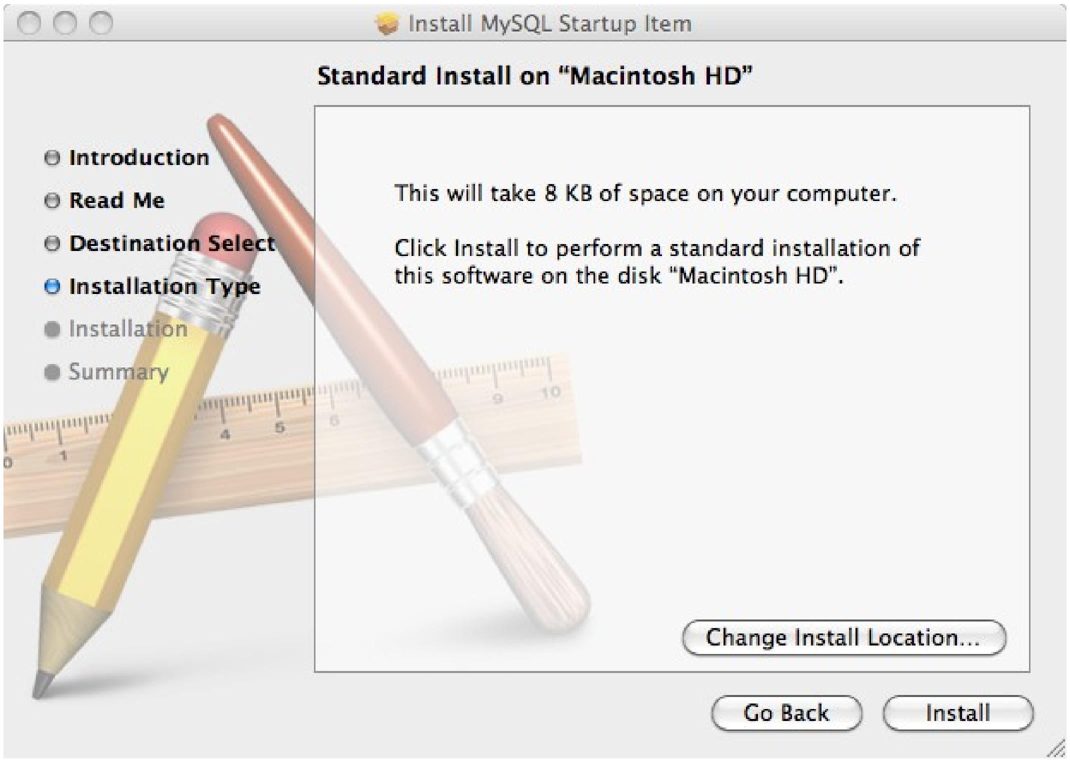


Select a Destination

Select the disk where you want to install the MySQL Startup Item software.

36.51 GB free
159.7 GB total

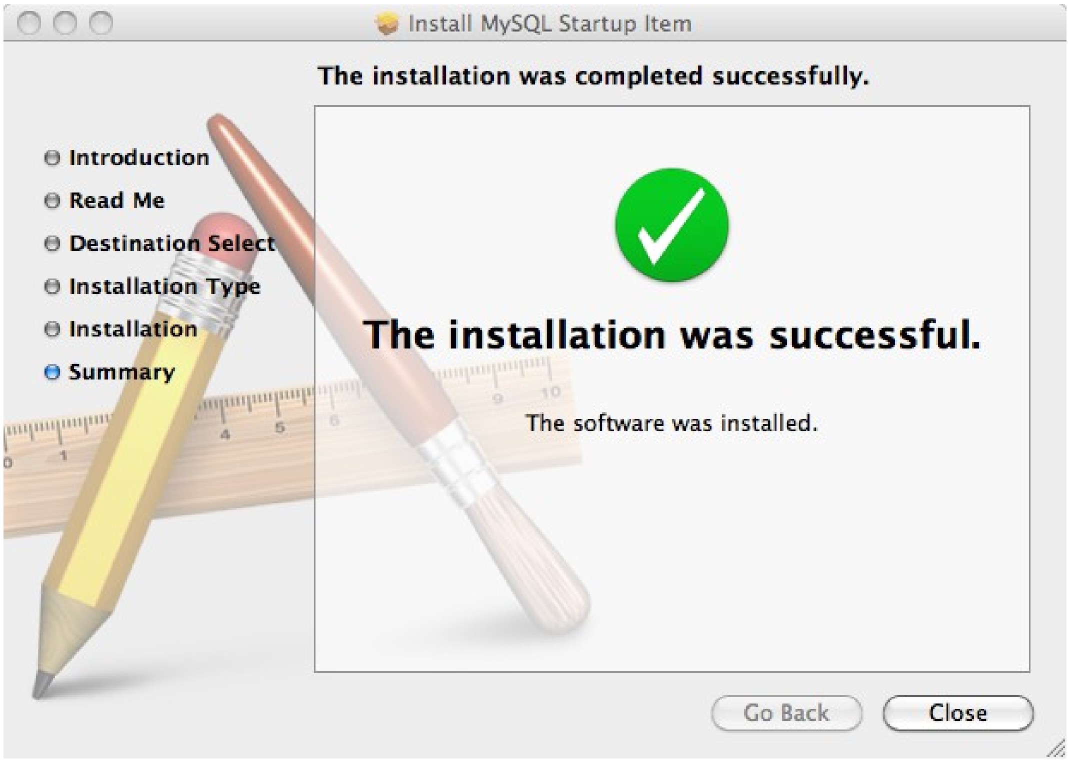
Installing this software requires 8 KB of space.



Standard Install on “Macintosh HD”

This will take 8 KB of space on your computer.

Click Install to perform a standard installation of this software on the disk “Macintosh HD”.



The installation was completed successfully.

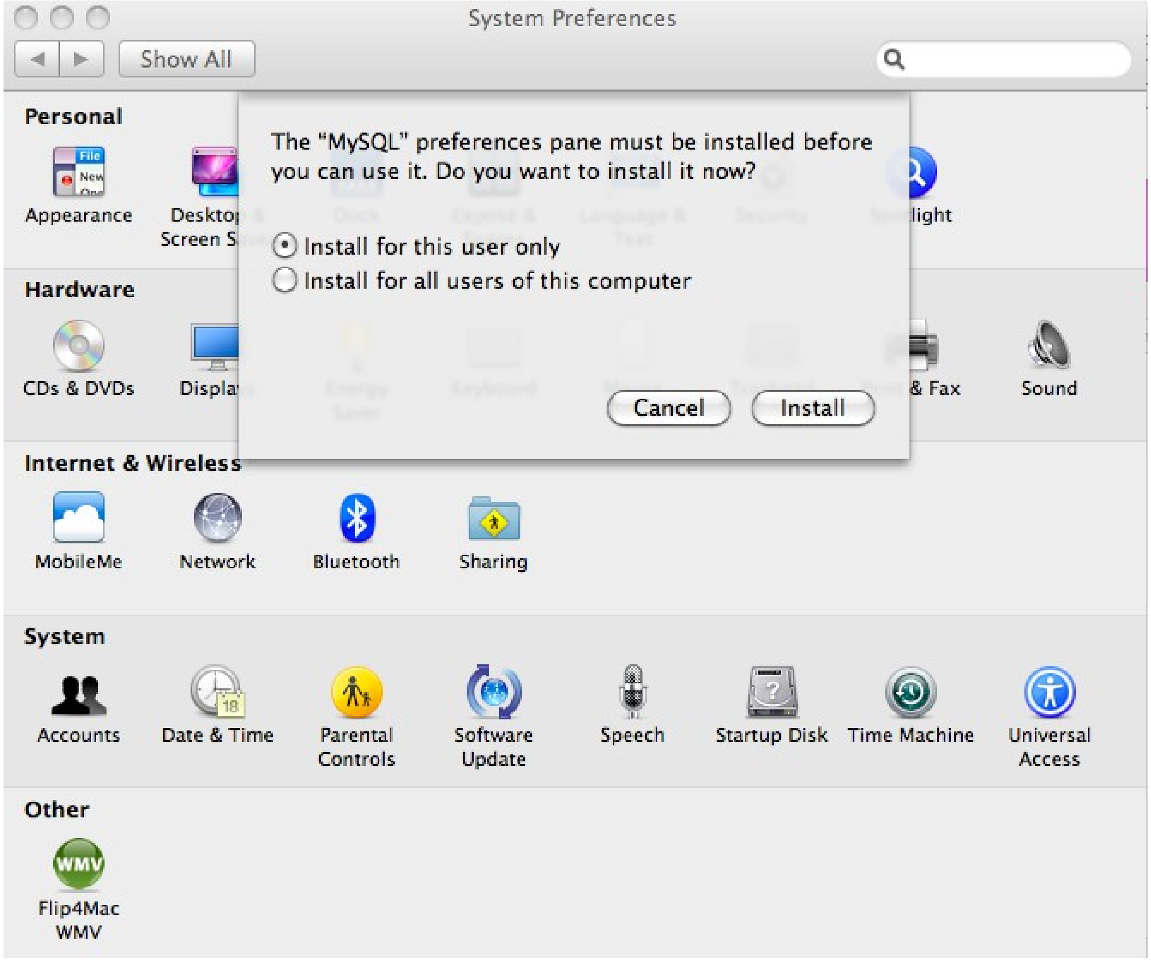
The installation was successful.

The software was installed.

Launch the MySQL Preferences Pane installer.  This will install MySQL pane in System Preferences.

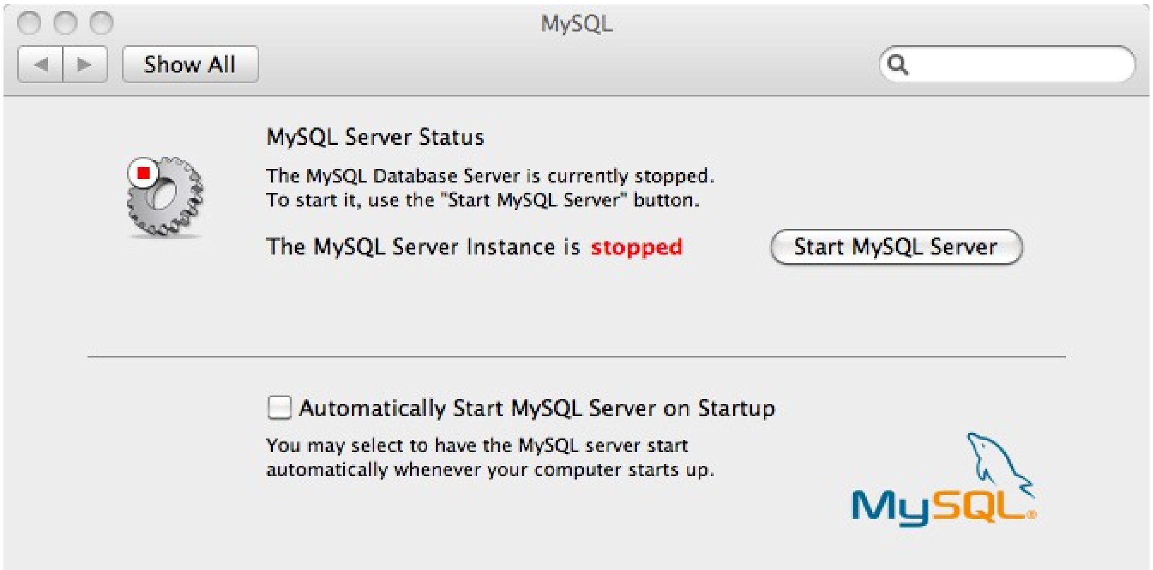
Click on the install button

Click on Start MySQL Server button and check the checkbox to auto start MySQL on boot.



The "MySQL" preferences pane must be installed before you can use it. Do you want to install it now?

● Install for this user only
○ Install for all users of this computer



MySQL Server Status

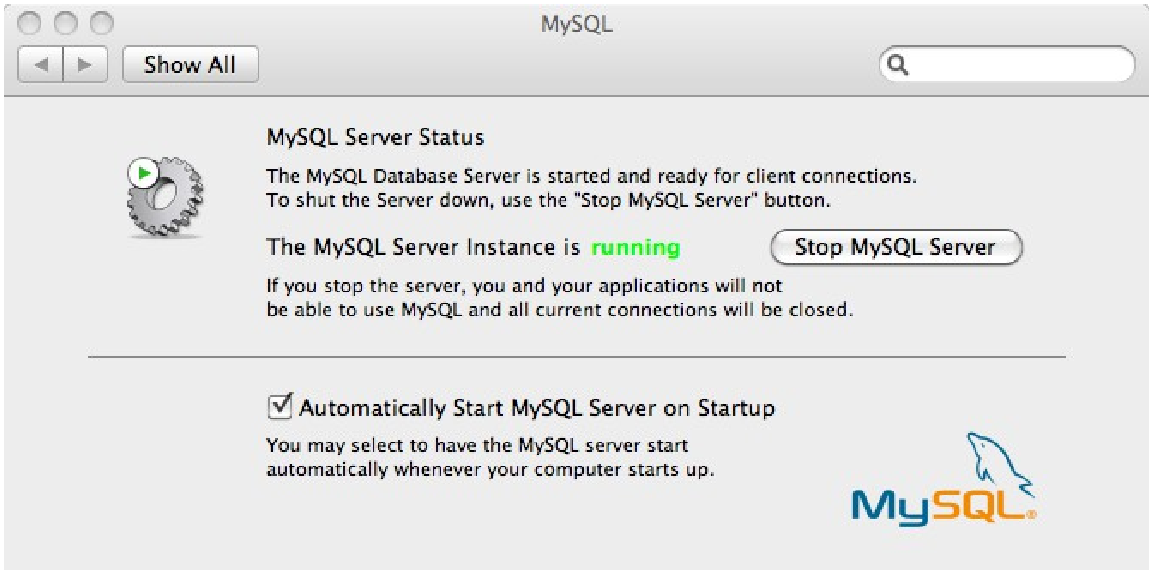
The MySQL Database Server is currently stopped.
To start it, use the "Start MySQL Server" button.

The MySQL Server Instance is **stopped**

□ Automatically Start MySQL Server on Startup

You may select to have the MySQL server start automatically whenever your computer starts up.

Everything looks good.



MySQL Server Status

The MySQL Database Server is started and ready for client connections.
To shut the Server down, use the "Stop MySQL Server" button.

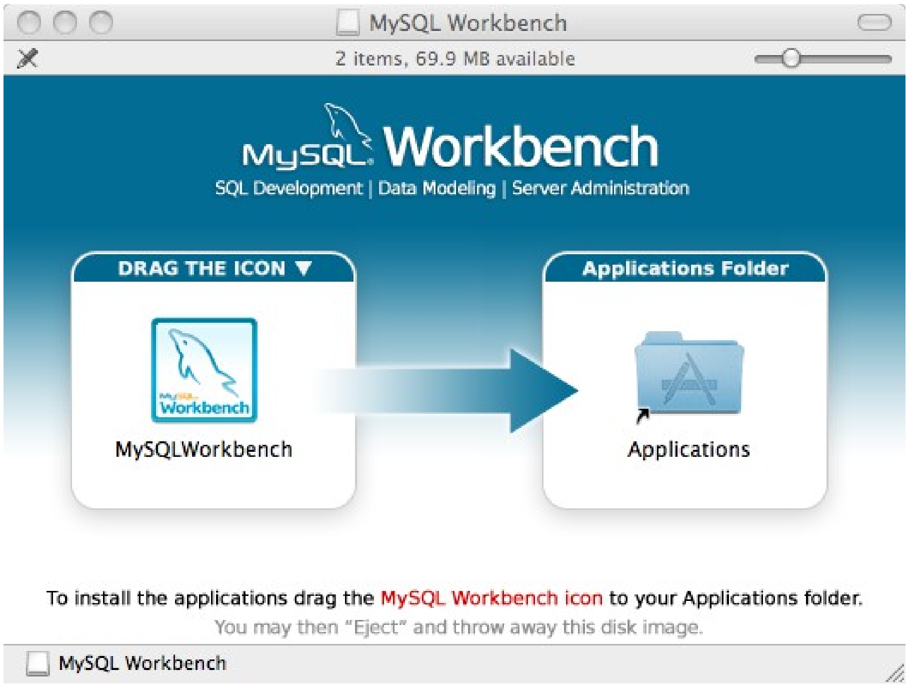
The MySQL Server Instance is running

If you stop the server, you and your applications will not
be able to use MySQL and all current connections will be closed.

☑ Automatically Start MySQL Server on Startup

You may select to have the MySQL server start
automatically whenever your computer starts up.

Now download and install the MySQL Workbench for your platform.  This is a convenient GUI based interface to modify your tables and such. Or use your favorite table editor.



To install the applications drag the MySQL Workbench icon to your Applications folder.
You may then "Eject" and throw away this disk image.

Now we need to do the following to get to the point where there is a basic MySQL installation where we can create a database and tables and the Product Activation Server can access these tables.

To do this, open a terminal window and launch mysql from the command prompt.

Change the default 'root' password from blank to something more secure.  For this example, we will set the root'@'localhost' password to 'pickle'.  Note: When installing MySQL Server on Linux for example, the root@localhost password is set during the installation.  So you may or may not need to use the -p option when launching mysql.

You should add /usr/local/mysql/bin to your $PATH (OS X, Linix) or add the pathname to mysql.exe on Windows to your %PATH% environment variable.  Or lastly just type the full pathname to the mysql executable.

```
$ /usr/local/mysql/bin/mysql -u root -p
Enter Password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.5.15 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('pickle');
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

While we are still logged in as 'root', so lets go ahead and create the 8 tables that the Product Activation Server uses.

Note: the 'source' command takes a pathname to the create.sql file which is located in the 'bin' folder of the pspro_<plat> installation folder.  Substitute the correct pathname for your installation folder.

The create.sql file is located int the 'bin' folder of your installation.  It creates all the necessary tables and adds some sample records which you can use as an example.

```
mysql> source create.sql

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.11 sec)

Query OK, 0 rows affected (0.10 sec)

Query OK, 0 rows affected (0.11 sec)

Query OK, 0 rows affected (0.10 sec)

Query OK, 0 rows affected (0.10 sec)

mysql>
```

Now we create the username/password pair for the Product Activation Server to login to the MySQL database. The username we will create is 'pasadmin' and set the password to 'secret'. You can set either one of these to your choosing.

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create user 'pasadmin'@'localhost' identified by 'secret';
Query OK, 0 rows affected (0.00 sec)
mysql> grant all privileges on sgpadb.* to pasadmin@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> quit
$
```

Now we need to edit the Product Activation Server 'control.txt' file which it uses on startup to gather information about the MySQL database connection, tables, usernames, etc.  This file is located in the 'bin' folder.

Control.txt file:

```
#
# Control file for the product activation server
#

SERVER=localhost PORT=29775
# the hash value for license generation can be either rc4 or sha256
CERTIFICATE_HASH_METHOD=rc4
SERVER_PASS=xyzzy
MYSQL_HOST=127.0.0.1
MYSQL_PORT=3306
MYSQL_USER=pasadmin
MYSQL_PASS=secret
MYSQL_DBNAME=sgpadb
MYSQL_DEMO_NODE_TABLE=demo_node
MYSQL_DEMO_NODE_DATA_TABLE=demo_node_data
MYSQL_DEMO_FLOAT_TABLE=demo_float
MYSQL_DEMO_FLOAT_DATA_TABLE=demo_float_data

MYSQL_PAID_NODE_TABLE=paid_node
MYSQL_PAID_NODE_DATA_TABLE=paid_node_data
MYSQL_PAID_FLOAT_TABLE=paid_float
MYSQL_PAID_FLOAT_DATA_TABLE=paid_float_data
```

The SERVER_PASS=  should be set to the password required for a user to remotely manipualte the Product Activation Server using the 'sgpautil' application.

The MYSQL_HOST= should be set to 127.0.0.1 if the MySQL Server is run on this machine.  Otherwise, it should be set to the hostname where the MySQL Server is running.

The MYSQL_PORT= allows you to specify the MySQL port number when running MySQL on a remote host.

The MYSQL_USER= is set to the username we gave access to the PAS database tables.

The MYSQL_PASS= is the password we set for the pasadmin user.

The MYSQL_DBNAME= is the name of the sgpadb database.

The remaining table names are correct for the server to operate.

## Starting the Product Activation Server

The Product Activation Server can be started from within a terminal window or as a Service on Windows. The sgpautil application is used to install or remove the Windows Service. For the other platforms, the service can be run in the background or for demonstration purposes using the -b option.

To launch the Product Activation Server from a terminal window, the following example shows this process.

```
$ cd pspro_mac64/bin
$ sgpad
Usage: sgpad {-b} -v <pa_server> -f <control_file> {-d <debug_log_file>}

$ sgpad -v sgpaserver -f control.txt
 sgpad release 5.0 on 1-mar-2013
 2014/3/1 09:53:52 Server starting on localhost port 29775
 Ready...
```

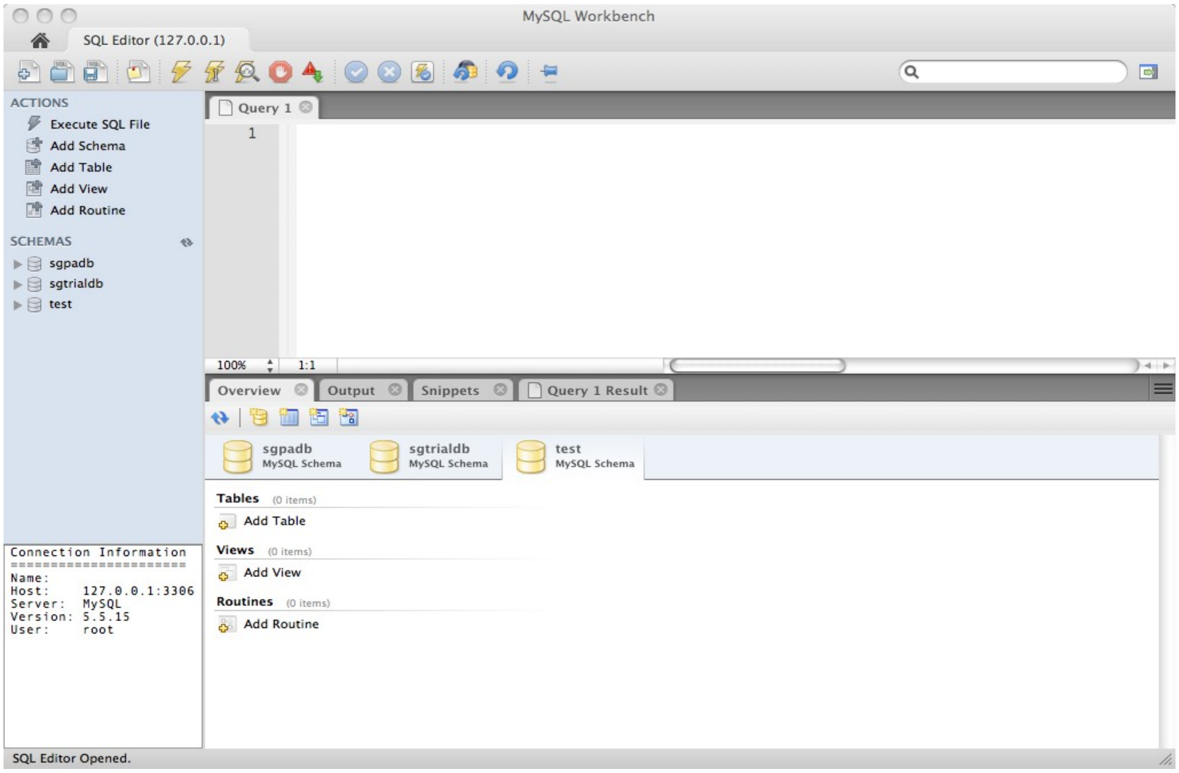
The Product Activation Server is now running and waiting for requests on TCP/IP port 29775. But before we can make requests for products to be activated, we have to instruct the MySQL database which products we want to activate as well as for how long, and also control other aspects of the activation process. To do this, we will use the MySQL Workbench to populate the required rows in the one of the tables.

For this example, we will enable the PAS to generate “demo” node-locked licenses for our test product named “monitor” version “5.0”. We will allow a 30 day trial license to be generated and all requests for demo licenses will be locked to the Macintosh Serial Number for Macintosh computers, and the ethernet address for computers running Windows. Since our imaginary product only runs on OS X and Windows, we only need to specify the locking attribute for these two.

To get started, launch the MySQL Workbench and click on “Open Connection to Start Querying”. Then click “Ok”. You will be prompted to enter the 'root'@'localhost' password twice. We set it to 'pickle'.

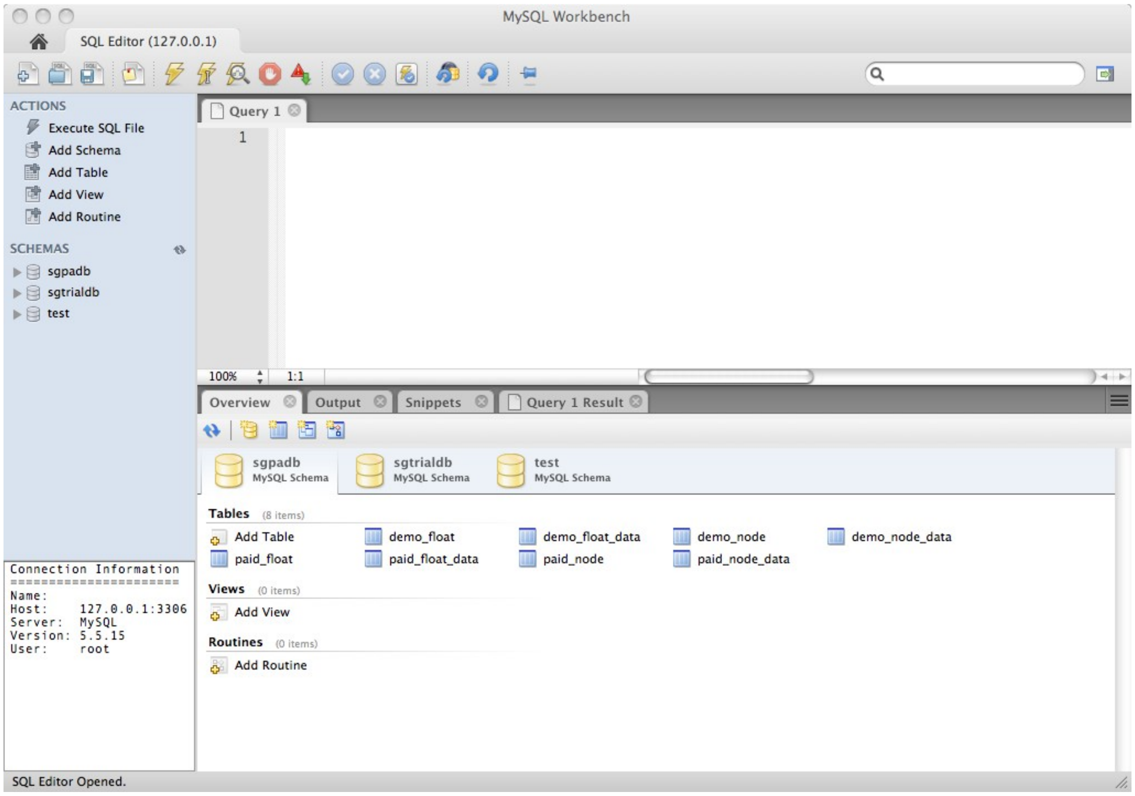
Once you have done this, you will see a screen like the following.

**Note: The Demo edition of SafeGuard LM ONLY supports the “any” host ID. Do not attempt to use any other host ID with the demo edition, as you will receive the error “Not licensed for ….” error code.**



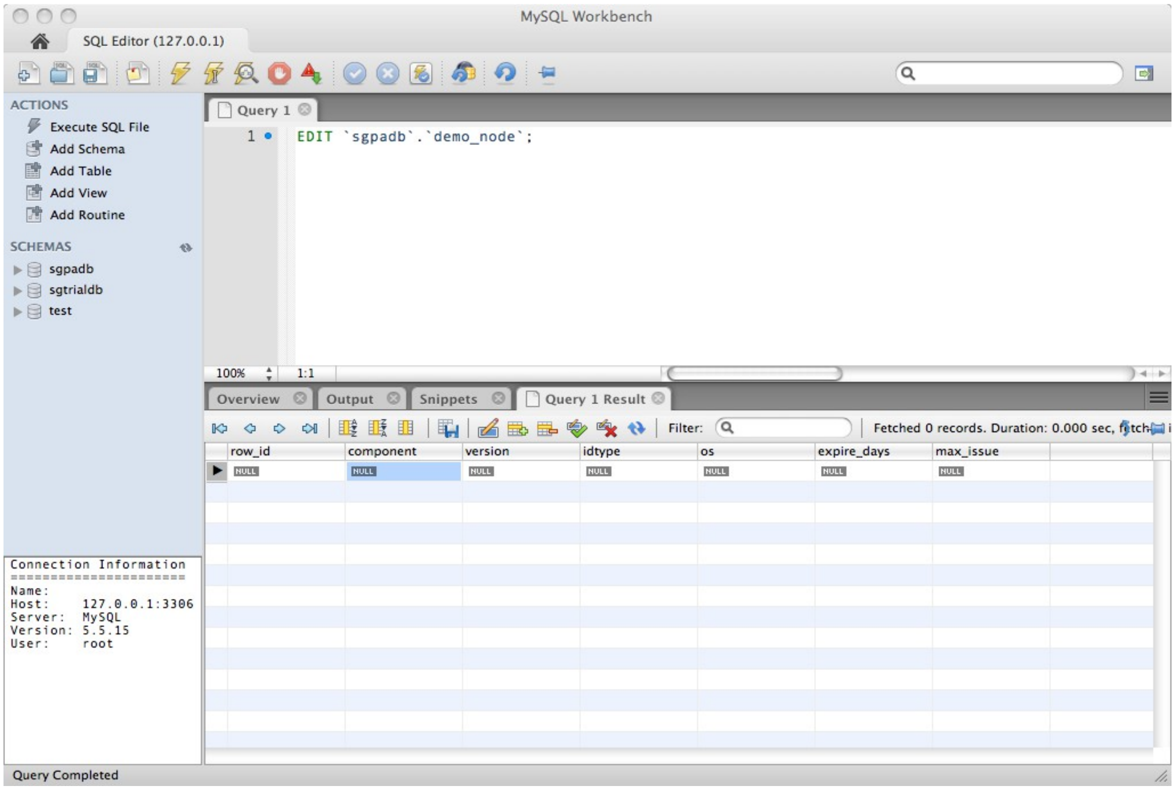
```
Connection Information
======================
Name:
Host:    127.0.0.1:3306
Server:  MySQL
Version: 5.5.15
User:    root
```

Click on the “sgpadb” icon in the “Overview” tab to display all eight tables created earlier.  You should now see all eight tables.



```
Connection Information
======================
Name:
Host:    127.0.0.1:3306
Server:  MySQL
Version: 5.5.15
User:    root
```

Click on the “demo_node” table. This is the table that controls the demo node-locked license creation for your product “monitor”. Here we will populate two rows. One for license creation attributes for our OS X users, and the other for our Windows users.



```
1 • EDIT `sgpadb`.`demo_node`;
```

Fetched 0 records. Duration: 0.000 sec, fetch

| row_id | component | version | idtype | os | expire_days | max_issue |
|---|---|---|---|---|---|---|
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

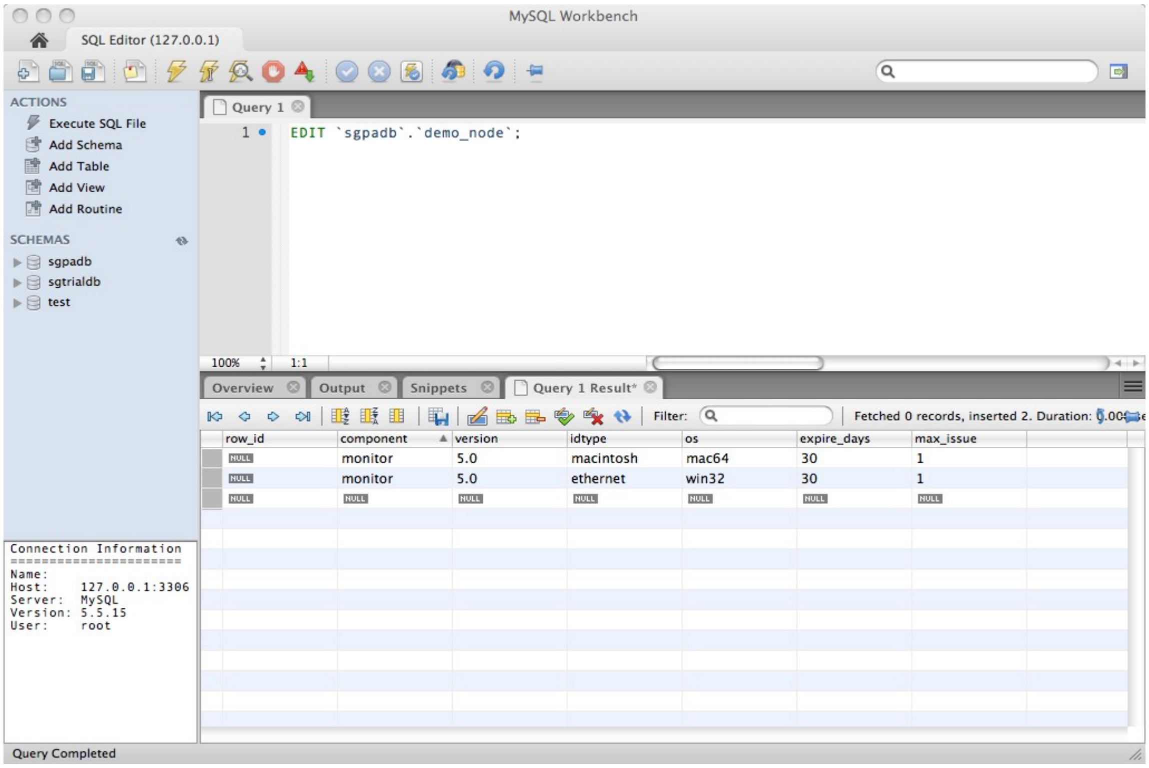
```
Connection Information
======================
Name:
Host:     127.0.0.1:3306
Server:   MySQL
Version:  5.5.15
User:     root
```

Here is a description of the seven columns.

row_id — This is automatically populated and does not need to be set.
component — This contains the name of our test product “monitor”
version — This is the version number of our test product “5.0”
idtype — The locking attribute for the “monitor” test product
os — Which operating system these attributes are applied to
expire_days — How many days from when the demo license is requested will it expire (-1 means “never”)
max_issue — How many times the demo license can be re-generated for a particular user

Now we will populate this table with our “monitor” product for our two supported platforms. Platforms are specified as a string with the following possible choices. mac32, mac64, win32, win64, lin32, lin64. Idtype's are specified as string values of the **SG_IDTYPE** enum's located in the lib/psprolib_enum.h file. These are the possible locking attributes available. Not all idtypes are available on each platform. For example, “macintosh” is not available on Windows. That is why we specify “ethernet” for Windows. This is just to emphasize the fact that we can control license generation by platform. See the DeveloperGuide.pdf for your platform to see which locking attributes are available.

**Important:** Once we add the two rows, we must commit the data to the table. To do this, under the Query 1 Result tab, click on the “apply changes to data” (green) icon, and then press the Apply button to apply the changes. See the following screens.



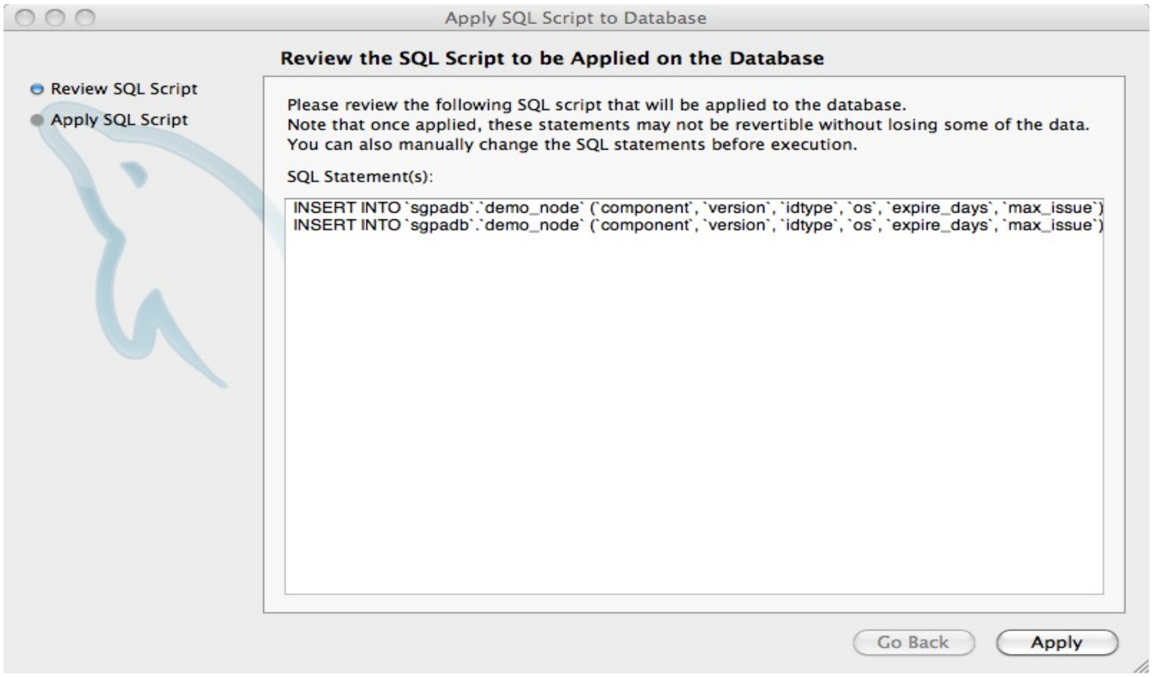
```
EDIT `sgpadb`.`demo_node`;
```

Fetched 0 records, inserted 2. Duration: 0.00

| row_id | component | version | idtype | os | expire_days | max_issue |
|---|---|---|---|---|---|---|
| NULL | monitor | 5.0 | macintosh | mac64 | 30 | 1 |
| NULL | monitor | 5.0 | ethernet | win32 | 30 | 1 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```
Connection Information
======================
Name:
Host:    127.0.0.1:3306
Server:  MySQL
Version: 5.5.15
User:    root
```

Query Completed



Review the SQL Script to be Applied on the Database

Please review the following SQL script that will be applied to the database.
Note that once applied, these statements may not be revertible without losing some of the data.
You can also manually change the SQL statements before execution.

SQL Statement(s):

```
INSERT INTO `sgpadb`.`demo_node` (`component`, `version`, `idtype`, `os`, `expire_days`, `max_issue`)
INSERT INTO `sgpadb`.`demo_node` (`component`, `version`, `idtype`, `os`, `expire_days`, `max_issue`)
```

The Product Activation Server is now ready to issue 30 day demo licenses for the product “monitor” for OS X and Windows.  The “monitor” license can be issued only once for any one machine requesting a license.

At this point, we can test the PAS using an example C application. Product activation can be done with C, Java or Python. The following is a simple example of activating a “trial” product over the Internet once.  The steps involved are first to see if we already are licensed for the “monitor” product.  If we are, then continue to execute.  Otherwise, activate the product and allow the application to run for 30 days from the activation date. During the trial period, the user could purchase your product and your product could activate itself permanently after a similar record is added to the “paid_node” table.  More on that later.

The following example C program is simplified for the purpose of example.  More error checking, and many more options can be performed.  The basic function of this sample is to show that first the “monitor” application can be authorized over the Internet.

The following C program (pas.c) is located under the examples/c installation folder for your convenience. To build it, run the appropriate build shell script for your platform.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "../../h/psprolib.h"
main(int nargs, char *args[])
{

  int retVal, daysRemaining;
  FILE *fp = (FILE *)0;
  char szMessage[1024];
  char outsrv[256],outlic[512];

  if (sgAuthorized("@pas.lic","monitor","5.0") != SG_AUTHORIZED) {
    // not authorized yet, so contact the PAS server for a demo license
    retVal = sgActivate("www.yourcompany.com",
            29775,SG_ACTIVATE_DEMO_NODE,
            NULL, "monitor", "5.0", NULL, outsrv, outlic);
    if (retVal == SG_SUCCESS) {
      // install the new license side-by-side to this application
      sgEnableComponentLine("@pas.lic",outlic);
      fprintf(stdout,"Activated the \"monitor\" application....\n");
    } else {
      sgGetLastErrorString(szMessage);
      fprintf(stdout,"%s\n",szMessage);
      exit(1);
    }
  }
  sgExpireDays("@pas.lic","monitor","5.0",&daysRemaining);
  fprintf(stdout,"Running the \"monitor\" application for %d more
days....\n",daysRemaining);
  exit(0);
}
```

Build the pas.c application and run it from a terminal window. You will see that the first time it is executed, it obtains a license over the Internet from the Product Activation Server. The second time, after the license has been installed locally, the application runs without contacting the activation server.

```
$ cd pspro_mac64/examples/c
$ source buildmac64.sh
…
$ ./pas
Activated the "monitor" application....
Running the "monitor" application for 30 more days....
$ ./pas
Running the "monitor" application for 30 more days....
```

Here is the output of the Product Activation server after the “pas” application was executed. You can see a demo-node license was requested, and provided to the client that will expire in 30 days.

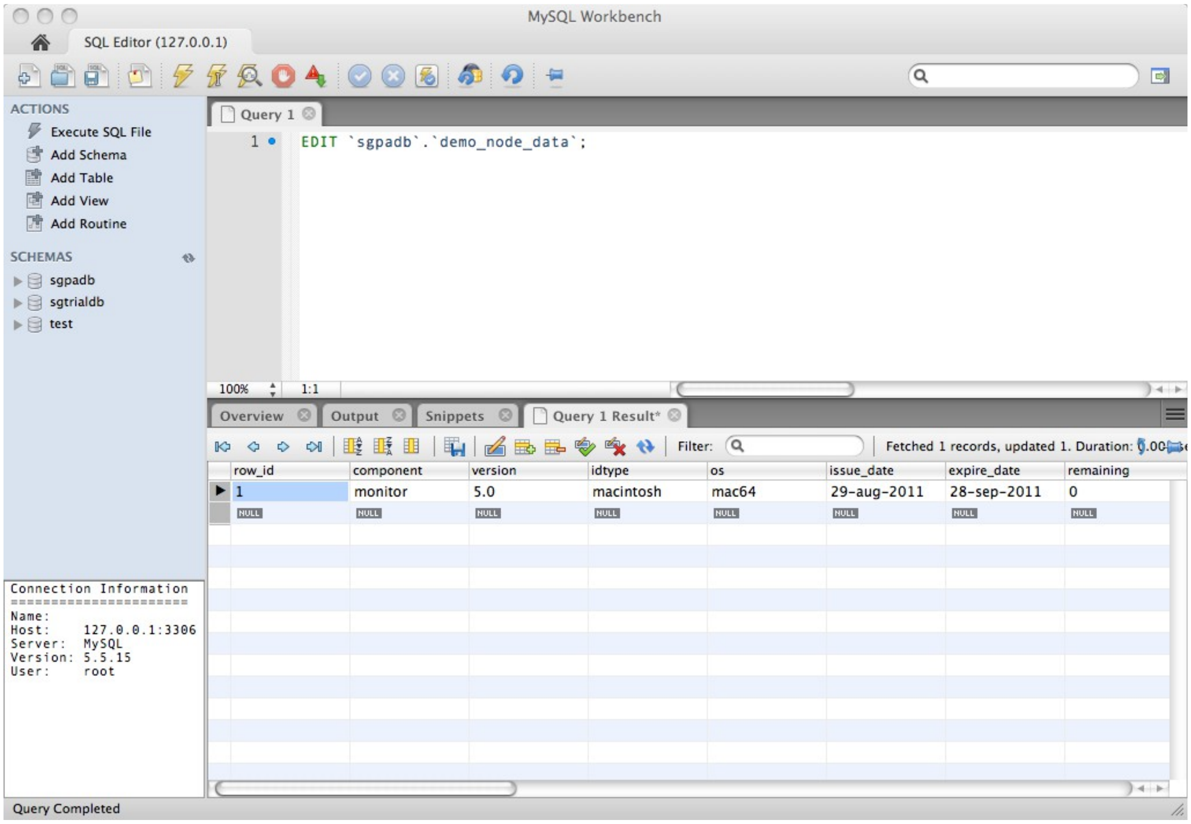
```
$ cd pspro_mac/bin
$ sgpad
Usage: sgpad {-b} -v <pa_server> -f <control_file> {-d <debug_log_file>}

$ sgpad -v sgpaserver -f control.txt
 sgpad release 5.0 on 1-mar-2014
 2014/3/1 09:53:52 Server starting on localhost port 29775
 Ready...
2014/3/1 11:44:45 CONNECT: USER: mark HOST: ps
 2014/3/1 11:44:45 REQUESTING: demo-node component 'monitor', version
'5.0', os 'mac64'
 2014/3/1 11:44:45 MySQL: Connecting...
 2014/3/1 11:44:45 MySQL: Connected...
 2014/3/1 11:44:45 ISSUING: demo-node license for component 'monitor',
version '5.0', os 'mac64'
 2014/3/1 11:44:45 LICENSE: NAME=monitor VERS=5.0 ID=w8814vhr0p1
IDTYPE=macintosh EXPIRES=15-jan-2015
CERT=fe81a69208f5a84cc88162c9df
 2014/3/1 11:44:46 DISCONNECT: USER: mark HOST: ps
```

Now lets look at the MySQL table "demo_node_data" table. It should contain a record for the requested platform. The record contains much more information from the client machine. Notice the expiration date and the remaining times the license can be issued. Also, other locking attributes are gathered like the ethernet address, Macintosh serial number, and other things.

The fact that we told the "demo_node" table that if a OS X machine requests a demo "monitor" license, we would want it locked to the Macintosh Serial Number. And in fact, if you look at the output from the PAS, you will see that the license generated was indeed locked to that serial number because it was a Mac that requested the license.

Here is a couple of screen shots from the "demo_node_data" table



```
EDIT `sgpadb`.`demo_node_data`;
```

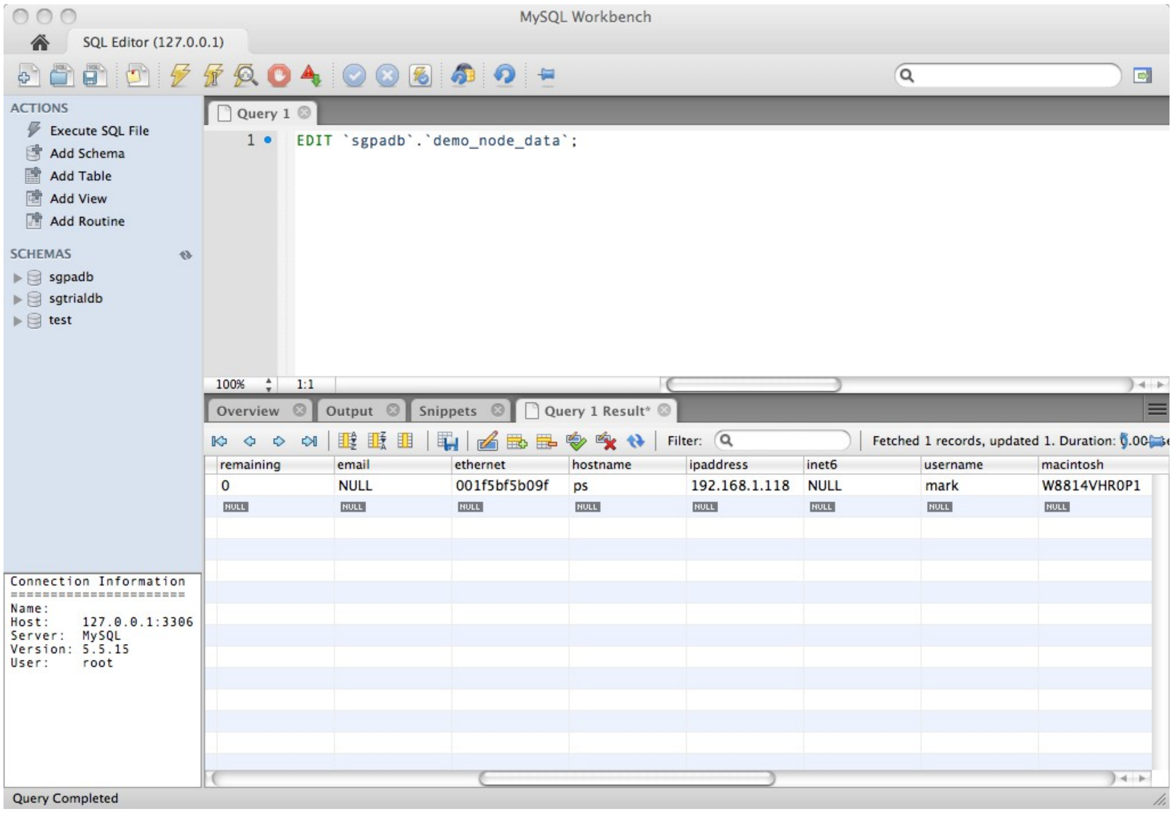
Fetched 1 records, updated 1. Duration: 0.00

| row_id | component | version | idtype | os | issue_date | expire_date | remaining |
|---|---|---|---|---|---|---|---|
| 1 | monitor | 5.0 | macintosh | mac64 | 29-aug-2011 | 28-sep-2011 | 0 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```
Connection Information
======================
Name:
Host:     127.0.0.1:3306
Server:   MySQL
Version:  5.5.15
User:     root
```

Now that we have that client machine registered and possibly others, we can perform queries on the database table and generate reports. Attributes like the email address, are sent only if you ask it from the user and then pass it on. This would be part of your "registration" process when the user installs the software the first time.

Other attributes like the "issue_date" are populated when the user actually installed and registered the software. If by chance you wanted to extend that particular users trial period, all you would have to do is change the date in the "expire_date" field. Once the original timeout occurred, the license check would fail, and the PAS would be contacted for the new license. You would also have to increment the "remaining" field to 1 so that the license could be once more generated.



```
EDIT `sgpadb`.`demo_node_data`;
```

Fetched 1 records, updated 1. Duration: 0.00

| remaining | email | ethernet | hostname | ipaddress | inet6 | username | macintosh |
|---|---|---|---|---|---|---|---|
| 0 | NULL | 001f5bf5b09f | ps | 192.168.1.118 | NULL | mark | W8814VHR0P1 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```
Connection Information
======================
Name:
Host:    127.0.0.1:3306
Server:  MySQL
Version: 5.5.15
User:    root
```

This is a basic installation of the Product Activation server for node-locked trial licenses. The basic process is the same for floating trial licenses.

For paid licenses, the “paid_node” table is setup about the same as the “demo_node” table with the difference that the “cust” field must be set to the ethernet address of the paid client. You can still lock the license to any of the other available idtypes, but the ethernet address is the key to issuing a paid license.

Refer to the **DeveloperGuide.pdf** for more examples of idtypes for your platform.