

Windows Server 2012 - Where did my Start Menu go?

✘ As you've probably heard or seen for yourself, Microsoft has changed the look and feel of Windows Server 2012 in significant ways. Some will see this as positive change, and many (if not most) will regard it as a strange and difficult step to take towards a more obscure design.

Let's just step back for a minute and realize what the goal of the Microsoft Server 2012 interface was. Among the many reasons, Microsoft wanted to:

- Unify the look and feel with the Windows 8 and Azure "Metro" design
- Focus on tasks and repeatedly used items
- Provide more "one-click" delivery of programs
- Enhance the command line and scripting tools for better automation

So there is a bit of a dichotomy here. We have an interface with a more "web" feel to it, and at the same time we have more focus on scripting and command line options.

That being said, let's take a quick look. Firstly we land in the Server Manager screen in the same way that Server 2008 and 2008 R2 would launch by default.



As we can see, the common task items are found in the Server Manager screen now instead of having to reach them from the Start Menu and Control Panel where we found them in 2003 or 2008 systems.

In effect Microsoft tried to deliver the portal interface to reach your common tasks. Here is the Tools menu (formerly Administrative Tools) which is from the upper right of the Server Manager screen:



In the Manage menu we find the Roles and Features as well as the properties pages for Server Manager to customize the content you will find there. You can also add additional servers to manage here to provide a single interface for multiple devices.



As you navigate around the screen by hovering your mouse in the corners you will find different sidebar menus pop out. If you are looking for the Start Menu, you will find the Start Page option by hovering towards the lower left.



Ahh, that feels better ☐ If you want to add more items to the Start page you simply right click which shows the All Apps button:



As you can see there are plenty of applications available this way and each can be added as a Start

item or a Taskbar item with a simple right-click of any icon.



We will dive deeper into things with some future posts, but I wanted to get you warmed up with this. It's a big change to the look and feel, but if you think about it, it is a directional change and one that you will barely remember making in a year or so.

Shell Time

The common theme with Microsoft servers, applications and infrastructure systems is that PowerShell integration is the true tie amongst it all. Nearly every component in the Microsoft environment, from SharePoint to the Windows Server OS and from Azure cloud services to Lync, to SQL 2012 to IIS. It goes without saying that PowerShell is the go-forward choice to administer Microsoft environments.

And the PowerShell reach goes beyond Microsoft tools into many other application environments, most prominently with PowerCLI for VMware.

Just right-click the PowerShell icon to see what your initial options are and you will see the Run as Administrator, Run ISE as Administrator, Windows PowerShell ISE and Windows PowerShell.



I'm still a command line junkie so I prefer the raw shell. In order to do many of the things you need to do, you will have to run as Administrator.



As you can see the User Account Control is on by default so you will be prompted for many administrative functions to confirm that you are about to make a change.

Now we are in our familiar PowerShell command shell except this is the new and improved PowerShell 3.0 environment which makes it even more exciting!



Let's leave it there for now and I encourage you to put together a lab machine and start clicking around to find all the new ways to do old things, and more importantly to embrace the new ways to do new things. Enjoy!

[MySQL - Create database, user and privileges](#)



This is a tip for the new MySQL users out there. A common and dangerous practice used by many new developers and admins is to use the root account to access databases. While this certainly gives

you the access you need to manage your databases and data from your applications it also elevates privileges far beyond what you really want.

Let's assume that you are creating a database called **mycontent** for an application. You will also want to create a user that has access to just that database so that it isolates the rights to only the content you want to affect.

Decide on a good name for a user. For this example I will use the classic format of `dbname_user` so our user will be **mycontent_user** which is what we will use for our application.

Make sure you choose a complex password as well. It is always best to combine upper and lower case as well as numbers. Be careful with special characters because using some characters like `/%*` can cause issues with systems when they pass the characters to a command. We are going to use **7U2JkyGg9TwbNnQ** as our example password.

Launch MySQL from the command line as a user who has privileges to create databases and manage security.

```
mysql -u myadminusername -p
```

Type in your password at the 'Enter password:' prompt

Now we create the database:

```
create database mycontent;
```

Next we create your database user. I am assuming that your database server is also your web server so we will use the `@localhost` to define the user.

```
grant usage on *.* to mycontent_user@localhost identified by '7U2JkyGg9TwbNnQ';
```

Next step is to apply permissions to the database for your new user:

```
grant all privileges on mycontent.* to mycontent_user@localhost;
```

And finally we check our user is able to log in and access the database. Exit from your MySQL command line and log in again using the new account:

```
mysql -u mycontent_user -p'7U2JkyGg9TwbNnQ' mycontent
```

Welcome to the MySQL monitor. Commands end with ; or g.

Your MySQL connection id is 10

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.

Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.

Success!

Now you can use this account/password combination in your application and it will have all of the privileges necessary to manage the database, tables, indexes and content with much less chance of compromising your other databases in your system.

[Xcode installation on OSX Lion - iTunes error](#)

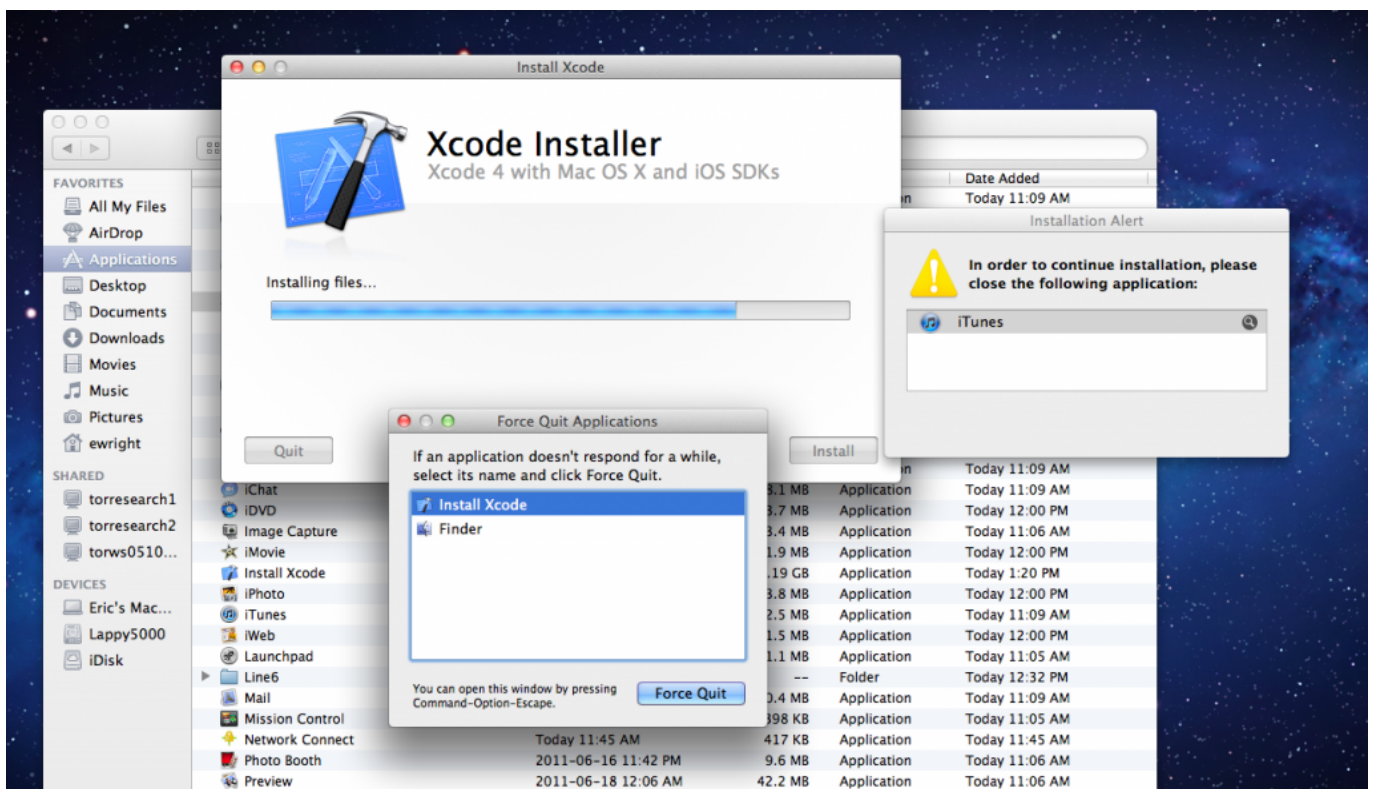
✘ Having trouble installing Xcode on Lion? Having just updated my Macbook to OSX Lion (yay!) I found that there was an odd issue with installing Xcode which you may have run into.

✘

First I downloaded the Xcode app from the App Store; a process that I'm still getting used to from my laptop OS. That being said, it is pretty convenient to download and manage apps this way.

Next step is to run the installation through the **Install Xcode** icon in the Applications folder. Then I walk away from the laptop for a while to let it work its magic only to find 10 minutes later that the installation stopped With the message "In order to continue installation, please close the following applications: iTunes".

Ok, but I don't have iTunes running. No really, I mean it. Not even minimized like a Windows guy and forgetting it's in the background, but really closed.



What iTunes?

Here is the catch. Although iTunes is not running, there is a process called iTunes Helper running in the background. This is the handy little tool that detects your devices and connects them.


The solution is simply to go to **Applications | Utilities | Activity Monitor** and within the process list, find and Quit the iTunes Helper process which is running in the background.



I see you hiding there

Now that you quit the process the Xcode installation will restart and complete without any further issues.

[The rebuild process - Recovering from my Macbook drive failure](#)

 You never see it coming, yet there it is staring you in the face. A perfectly frozen screen with no animation, no mouse movement, no keyboard response. The first thing you think is that it is just some errant process that ate up all of the resources and went off to chew on it in a tree somewhere.

I took the first logical step. Hard power down and boot back up. We've all done it a number of times before. This time was different though and I immediately had that sickening feeling. The Apple logo screen and it's little cutesy spinner were on the screen for around 5 minutes. My fears were realized soon after as the OS loaded after another 5 minutes only to hear the constant whirring of the hard drive and glacially slow response to mouse clicks in the UI.

Within 15 minutes the system had ground to a halt and multiple attempts to revive the poor overworked hard drive proved to be fruitless. I have been using Time Machine backups for

There are some points that I know will be troublesome:

- MySQL and Rails on Snow Leopard - 64 bit configuration was a pain last time, so it won't be any different this time
- Recovering and rebuilding iTunes library

Conversely, there are some things made very easy because of simple things:

- Rails and Ruby code - Not a problem because I use Unfuddle to keep all my code in a repository on the web
- Pictures - I learned from other's mistakes. I have a Time Machine backup from 6 weeks ago

- Documents and other tools - Thank you Google! I store a lot of items in Gmail and Google Docs for ease of use and ease of access. This has a doubling effect of saving me from this type of situation

Here is my build and how it all went together. It is more for my own personal notes, but if you ever wanted to setup a Macbook as a Ruby on Rails development environment as well as a pretty solid Garage Band recording studio, this is how you can do it.

1. Install Snow Leopard - my machine came with Leopard so I only have the core upgrade/install DVD
2. Install Xcode from the Snow Leopard DVD
3. Install Applications (iLife bundle) - I have to use the Leopard edition and update after the fact using Software Update
4. Install Google Chrome
5. Update all code to latest and greatest using Software Update
6. Install TextMate and configure as the default editor for the shell and for SVN (don't judge)

```
export EDITOR='mate -w'  
export SVN_EDITOR='mate -w'
```

7. Run the Rubygems updates

```
sudo gem update -system
```

8. Setup my code folder and get my active project code from Unfuddle
9. Go into each folder and run Bundle Install to get all the necessary components (Dear Rails 3.x, Thank you!)
10. Install MySQL (file: mysql-5.1.53-osx10.6-x86_64.dmg) including the package, Pref Pane and Startup Item
11. Configure MySQL (follow the MySQL section here: <http://weblog.rubyonrails.org/2009/8/30/upgrading-to-snow-leopard>)

Additionally I set the path include the MySQL bin folder by adding this line to my `~/.bash_profile` file

```
export PATH=/usr/local/mysql/bin:$PATH
```

12. Install and update MacPorts
13. Install uTorrent
14. Install my Line6 software. This is easy because I install a product called Line6 Monkey, plug in the TonePort UX2 and download and install the related software using Monkey

And that, as they say, is that. One fully functional Ruby on Rails environment and one fully

operational recording studio in a neat and tiny little aluminum case.