# Installing Cassandra in a Local Machine

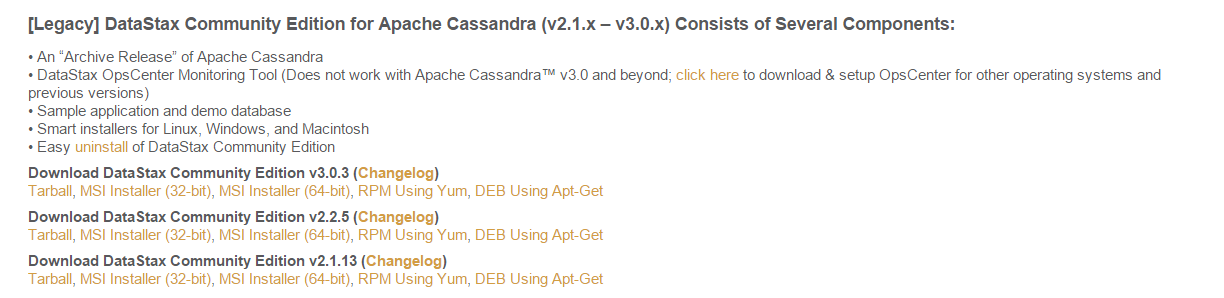
## Step 1: Download Installer

This document explains the steps for installing DataStax Cassandra Community Edition v2.1.13, the legacy version as the tested Java APIs work with this version.

Download the Installation package from the below site:

<http://www.planetcassandra.org/cassandra/>

Scroll down and go to section: DataStax Community Edition v2.1.13



OR

Windows 64 bit: <http://downloads.datastax.com/community/datastax-community-64bit_2.1.13.msi>

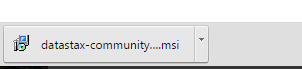
Windows 32 bit: <http://downloads.datastax.com/community/datastax-community-32bit_2.1.13.msi>

Tarball (Linux/Mac): <http://downloads.datastax.com/community/dsc-cassandra-2.1.13-bin.tar.gz>

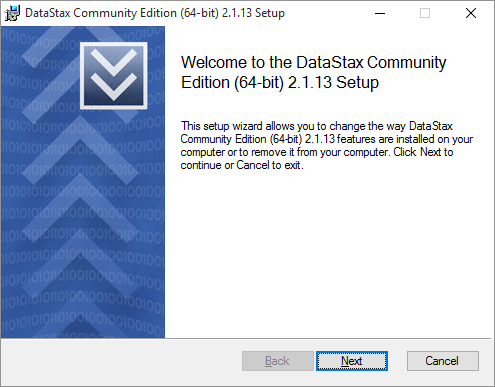
Below steps are explained for a Windows 64 bit machine.

Mac installation steps are given here: <http://www.datastax.com/2012/01/working-with-apache-cassandra-on-mac-os-x>

## Step 2: Run the Installer



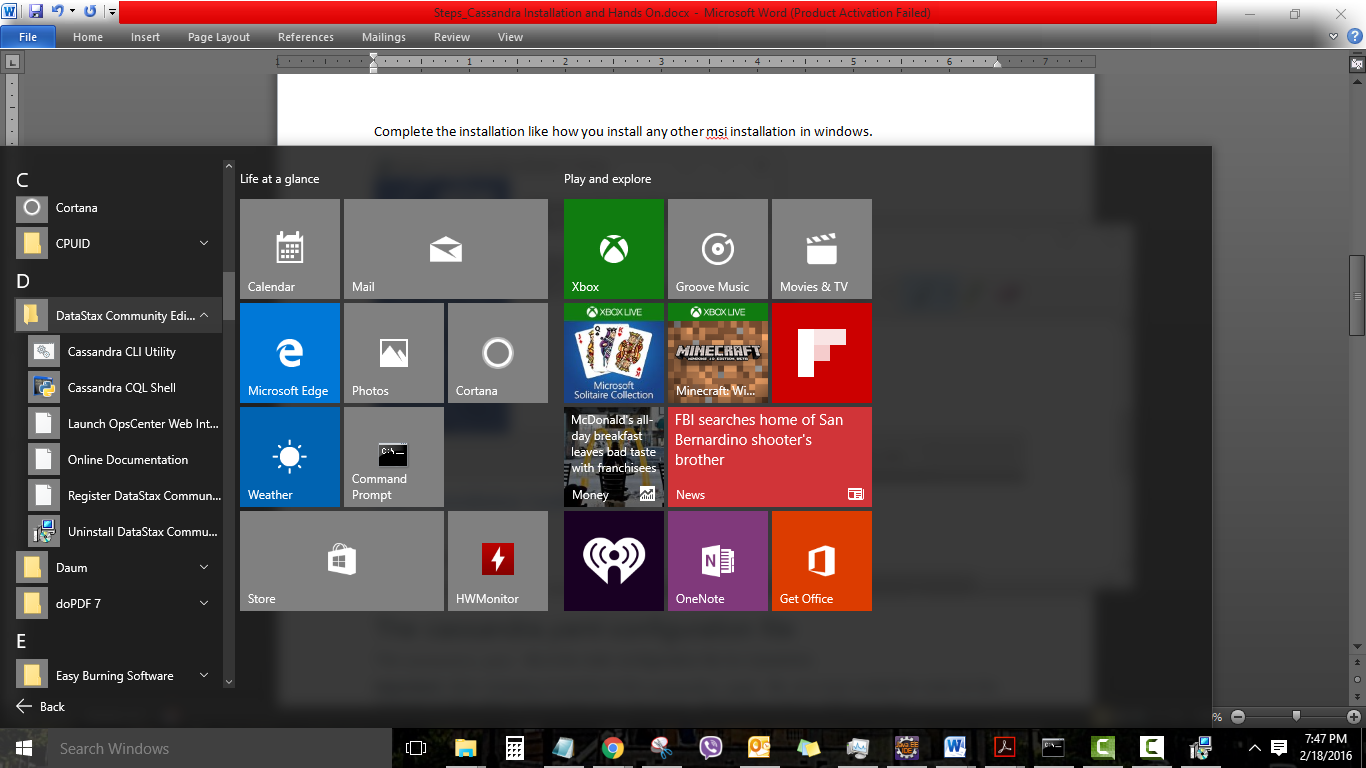
Complete the installation like how you install any other msi installation in windows.



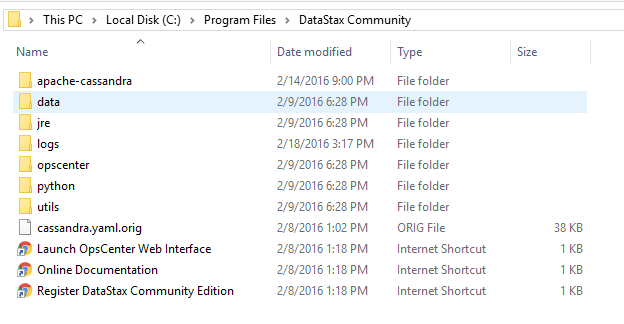
## Step 2: Installation Confirmation and Folder

The installation can be confirmed by checking the Programs Menu. The Cassandra will be installed under

C:\Program Files\DataStax Community (for a 64 bit normal installation)



Checking in Windows Explorer



## Optional Step: Nodetool Command

Binaries are located in the folder:

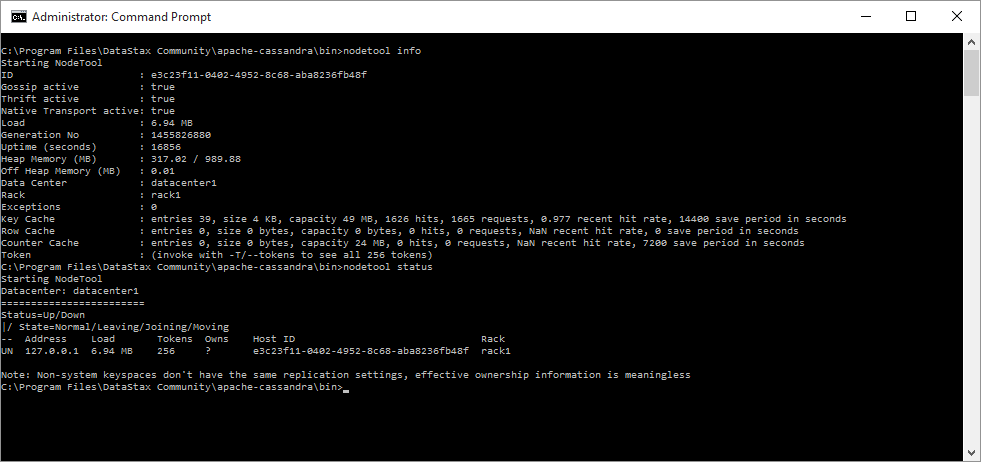
C:\Program Files\DataStax-DDC\apache-cassandra\bin

Open cmd command prompt (Run as administrator)

>cd C:\Program Files\DataStax Community\apache-cassandra\bin

>nodetool info

>nodetool status



## Optional Step: Check the cassandra.yaml configuration file

The cassandra.yaml file is the main configuration file for Cassandra.

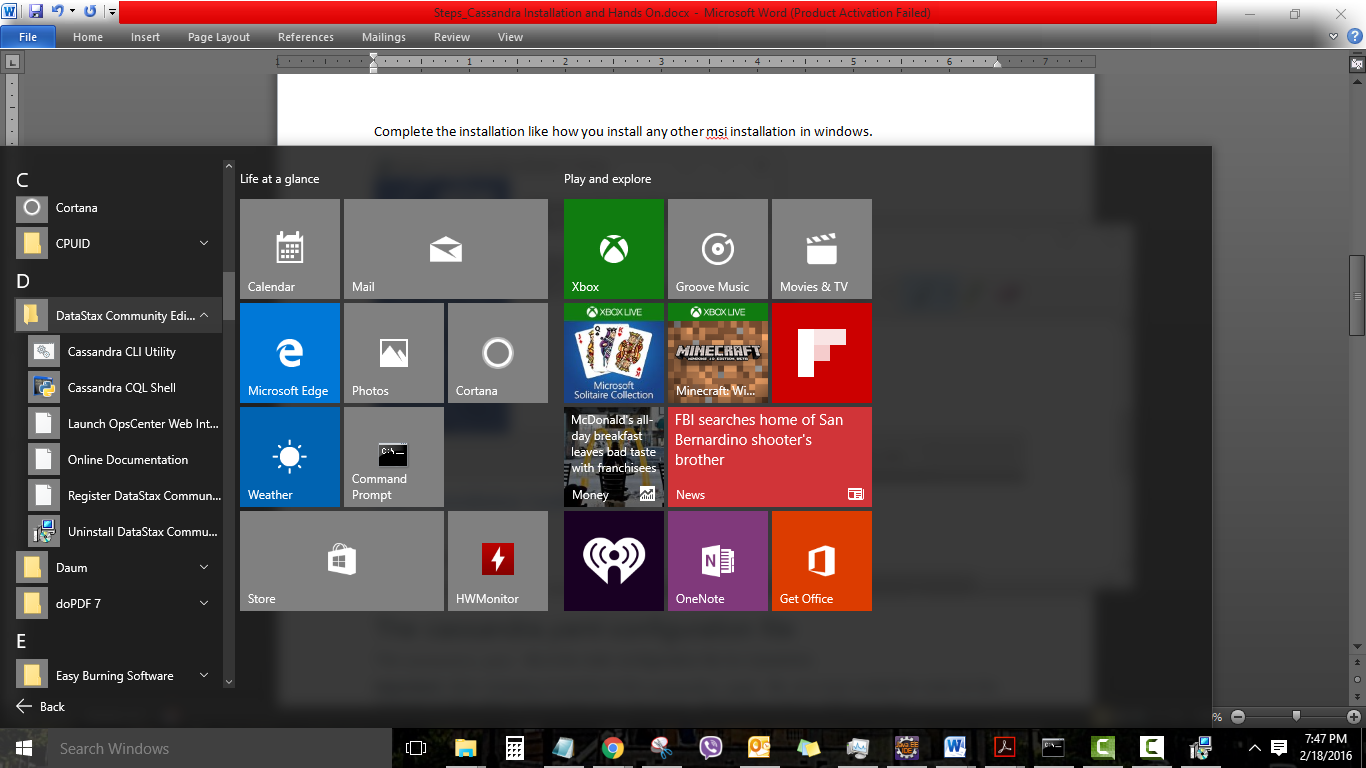
This is found in the location: C:\Program Files\DataStax Community\apache-cassandra\conf

**Important:** Not recommended to make changes in this file for this exercise.

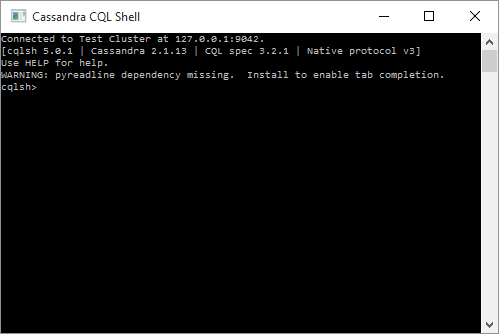
After changing properties in the **cassandra.yaml** file, you must restart the node for the changes to take effect.

## Step 3: CQL Shell

Launch Cassandra CQL Shell from the Programs Menu:



CQL Shell window opens up:



## CQL Command Samples

CREATE KEYSPACE **ecommerce** WITH REPLICATION = {'class' : 'SimpleStrategy', 'replication\_factor' : 3};

DESCRIBE KEYSPACE ecommerce;

USE **ecommerce**;

CREATE TABLE **products** (

**pdt\_id** int PRIMARY KEY,

**cat\_id** int,

**pdt\_name** varchar,

**pdt\_desc** varchar,

**price** float,

**shipping** varchar

);

USE **ecommerce**;

CREATE TABLE **category** (

**cat\_id** int PRIMARY KEY,

**cat\_name** varchar,

**cat\_desc** varchar

);

USE **ecommerce**;

INSERT INTO **products** (**pdt\_id**, **cat\_id**, **pdt\_name**, **pdt\_desc**, **price**, **shipping**) VALUES (006,105, 'Olay Regenerist Micro-Sculpting Cream Face', 'One 1.7 oz. jar of Olay face moisturizer', 19.79, 'Standard');

USE **ecommerce**;

INSERT INTO **products** (**pdt\_id**, **cat\_id**, **pdt\_name**, **pdt\_desc**, **price**, **shipping**) VALUES *(001,101, 'HP Envy Laptop', 'HP Laptop computer 17 inch, i7 processor, 16 GB, 1 TB, Intel Graphics', 899.00, 'Standard');*

USE **ecommerce**;

INSERT INTO **products** (**pdt\_id**, **cat\_id**, **pdt\_name**, **pdt\_desc**, **price**, **shipping**) VALUES *(002,101, 'Apple MacBook', 'Apple Laptop computer 13 inch screen, i3 processor, 4 GB, 1 TB, Intel Graphics', 1399.00, 'Standard');*

INSERT INTO **products** (**pdt\_id**, **cat\_id**, **pdt\_name**, **pdt\_desc**, **price**, **shipping**) VALUES *(005,104, 'Danby 0.7 cu. ft. Microwave Oven', 'Capacity of 0.7 cu. ft.10 different power levels', 54.00, 'Expedited');*

*//(001,101, 'HP Envy Laptop', 'HP Laptop computer 17 inch, i7 processor, 16 GB, 1 TB, Intel Graphics', 899.00, 'Standard');*

*//(002,101, 'Apple MacBook', 'Apple Laptop computer 13 inch screen, i3 processor, 4 GB, 1 TB, Intel Graphics', 1399.00, 'Standard');*

*//(003,102, 'Black & Decker Bread Toaster', 'Bread toaster for two slices', 21.00, 'Standard');*

*//(004,103, 'Learning Apache Cassandra - Paperback', '240 page paperback book on Cassandra. Author: J Moeller', 36.00, 'Expedited');*

*//(005,104, 'Danby 0.7 cu. ft. Microwave Oven', 'Capacity of 0.7 cu. ft.10 different power levels', 54.00, 'Expedited');*

*//(006,105, 'Olay Regenerist Micro-Sculpting Cream Face', 'One 1.7 oz. jar of Olay face moisturizer', 19.79, 'Standard');*

USE **ecommerce**;

INSERT INTO **category** (**cat\_id**, **cat\_name**, **cat\_desc**) VALUES (104, 'Beauty', 'Collection of Beauty and grooming');

*//(101, 'Computers and Laptops', 'Collection of PC, Laptops and NOtebooks');*

*//(102, 'Home & Kitchen', 'Collection of Home and Kitchen appliances and tools');*

*//(103, 'Books', 'Books and Literature');*

*//(104, 'Beauty', 'Collection of Beauty and grooming');*

# Cassandra - Java Integration

The below sections explains how to connect to Cassandra database to a Java application.

## Prerequisites

The prerequisites needed are:

* Running instance of locally installed Cassandra (One or more nodes)
* Java IDE like Eclipse.
* Cassandra Java APIs: Downloadable from <https://datastax.github.io/java-driver/2.1.9/>

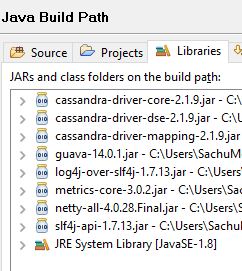
Or <https://datastax.github.io/java-driver/2.1.9/features/shaded_jar/>

* Other dependencies suggested jar files are: slf4j, netty-all, log4j, guava, metrics-core

## Create a new Java Project

Create a new Java project in Eclipse

Add the below libraries in the Java Build Path. (The jar files in perquisites should have been downloaded earlier)



## Java Code for basic Insert and select actions:

Create a new class in the project and use the below source code:

**package** casstest;

**import** com.datastax.driver.core.Cluster;

**import** com.datastax.driver.core.ResultSet;

**import** com.datastax.driver.core.Row;

**import** com.datastax.driver.core.Session;

**public** **class** cassandra\_tester {

**public** **static** **void** main(String[] args) {

System.***out***.print("\*\*\*\*\* Cassandra - Java Connection Tester \*\*\*\*\*\*");

Cluster cluster;

Session session;

// Connect to the cluster and keyspace "ecommerce" 127.0.0.1:9042/9160

cluster = Cluster.*builder*().addContactPoint("localhost").build();

session = cluster.connect("ecommerce");

// Use insert statement

System.***out***.println("Inserting into Cassandra Database...");

session.execute("INSERT INTO products (pdt\_id, cat\_id, pdt\_name, pdt\_desc, price, shipping) VALUES (005,104, 'Danby 0.7 cu. ft. Microwave Oven', 'Capacity of 0.7 cu. ft.10 different power levels', 54.00, 'Expedited')");

// Use select statement

String pdtid = **null**, pdtname = **null**, pdtdesc = **null**;

**float** price = 0;

// Use select statement

ResultSet results = session.execute("SELECT \* FROM products where pdt\_id = 5");

**for** (Row row : results) {

pdtid = Integer.*toString*(row.getInt("pdt\_id"));

pdtname = row.getString("pdt\_name");

pdtdesc = row.getString("pdt\_desc");

price = row.getFloat("price");

System.***out***.println("Product ID: " + pdtid);

System.***out***.println("Name: " + pdtname);

System.***out***.println("Description: " + pdtdesc);

System.***out***.println("Price: "+ price);

}

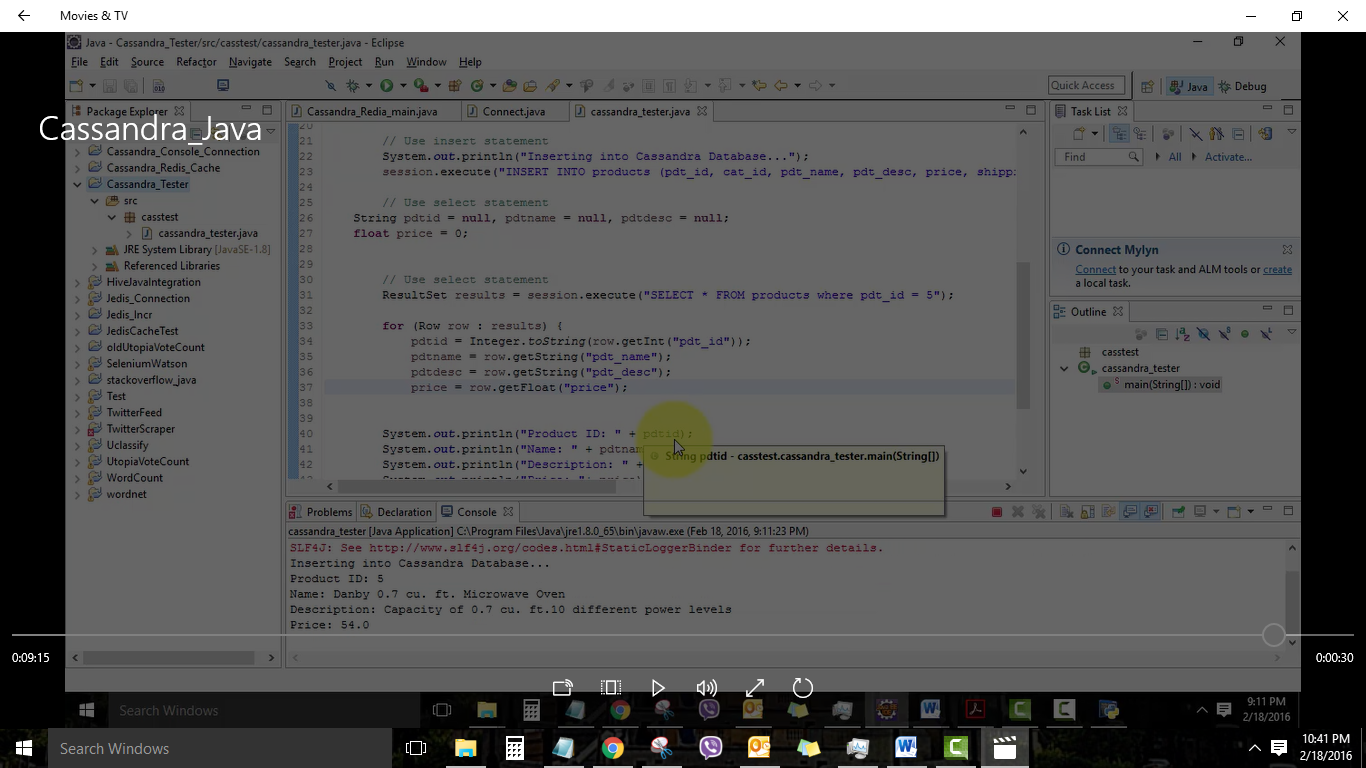
// Clean up the connection by closing it

cluster.close();

}

}

Results:



Additional Reading Suggested:

<https://academy.datastax.com/demos/getting-started-apache-cassandra-and-java-part-i>

<https://academy.datastax.com/demos/getting-started-apache-cassandra-and-java-part-ii>