Technologies used:

* JavaFX
* Mysql

JavaFX

JavaFX is a Java library using which you can develop Rich Internet Applications. By using Java technology, these applications have a browser penetration rate of 76%.

What is JavaFX?

JavaFX is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms. The applications developed using JavaFX can run on various devices such as Desktop Computers, Mobile Phones, TVs, Tablets, etc.

To develop **GUI Applications** using Java programming language, the programmers rely on libraries such as **Advanced Windowing Toolkit** and **Swing**. After the advent of JavaFX, these Java programmers can now develop GUI applications effectively with rich content.

Need for JavaFX

To develop **Client Side Applications** with rich features, the programmers used to depend on various libraries to add features such as Media, UI controls, Web, 2D and 3D, etc. JavaFX includes all these features in a single library. In addition to these, the developers can also access the existing features of a Java library such as **Swing**.

JavaFX provides a rich set of graphics and media API’s and it leverages the modern **Graphical Processing Unit** through hardware accelerated graphics. JavaFX also provides interfaces using which developers can combine graphics animation and UI control.

One can use JavaFX with JVM based technologies such as Java, Groovy and JRuby. If developers opt for JavaFX, there is no need to learn additional technologies, as prior knowledge of any of the above-mentioned technologies will be good enough to develop RIA’s using JavaFX.

Features of JavaFX

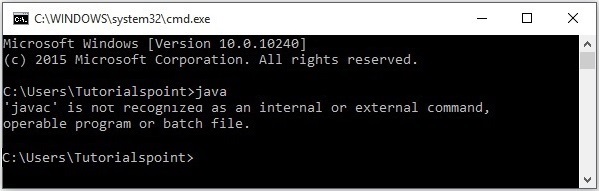
Following are some of the important features of JavaFX −

* **Written in Java** − The JavaFX library is written in Java and is available for the languages that can be executed on a JVM, which include − **Java, Groovy and JRuby**. These JavaFX applications are also platform independent.
* **FXML** − JavaFX features a language known as FXML, which is a HTML like declarative markup language. The sole purpose of this language is to define a user Interface.
* **Scene Builder** − JavaFX provides an application named Scene Builder. On integrating this application in IDE’s such as Eclipse and NetBeans, the users can access a drag and drop design interface, which is used to develop FXML applications (just like Swing Drag & Drop and DreamWeaver Applications).
* **Swing Interoperability** − In a JavaFX application, you can embed Swing content using the **Swing Node** class. Similarly, you can update the existing Swing applications with JavaFX features like embedded web content and rich graphics media.
* **Built-in UI controls** − JavaFX library caters UI controls using which we can develop a full-featured application.
* **CSS like Styling** − JavaFX provides a CSS like styling. By using this, you can improve the design of your application with a simple knowledge of CSS.
* **Canvas and Printing API** − JavaFX provides Canvas, an immediate mode style of rendering API. Within the package **javafx.scene.canvas** it holds a set of classes for canvas, using which we can draw directly within an area of the JavaFX scene. JavaFX also provides classes for Printing purposes in the package **javafx.print**.
* **Rich set of API’s** − JavaFX library provides a rich set of API’s to develop GUI applications, 2D and 3D graphics, etc. This set of API’s also includes capabilities of Java platform. Therefore, using this API, you can access the features of Java languages such as Generics, Annotations, Multithreading, and Lambda Expressions. The traditional Java Collections library was enhanced and concepts like observable lists and maps were included in it. Using these, the users can observe the changes in the data models.
* **Integrated Graphics library** − JavaFX provides classes for **2d** and **3d** graphics.
* **Graphics pipeline** − JavaFX supports graphics based on the Hardware-accelerated graphics pipeline known as Prism. When used with a supported Graphic Card or GPU it offers smooth graphics. In case the system does not support graphic card then prism defaults to the software rendering stack.

## Installing Java8

First of all, you will have to verify whether there is Java Installed in your system or not by opening the command prompt and typing the command “Java” in it.

If you haven’t installed Java in your system, the command prompt displays the message shown in the following screenshot.



Then install Java by following the steps given below.

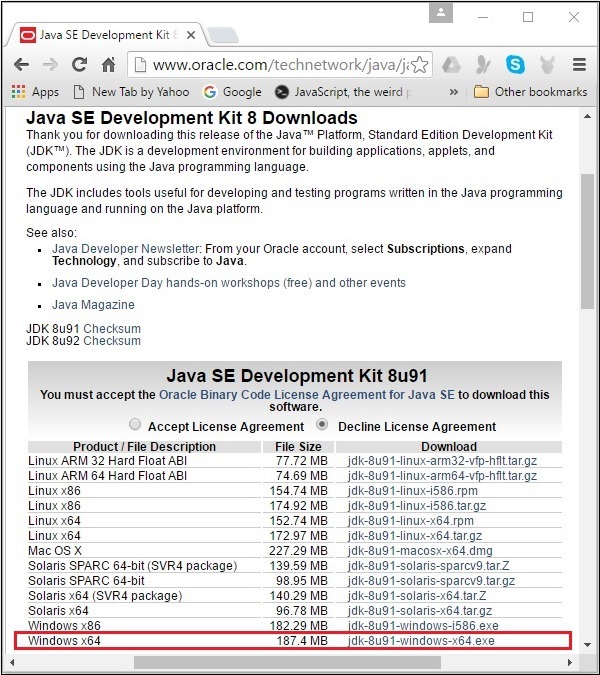
**Step 1** − Visit the [JavaSE Downloads](https://www.oracle.com/technetwork/java/javase/downloads/index.html) Page, click on the JDK **Download**button as highlighted in the following screenshot



**Step 2** − On clicking the Download button, you will be redirected to the **Java SE Development Kit 8 Downloads** page. This page provides you links of JDK for various platforms.

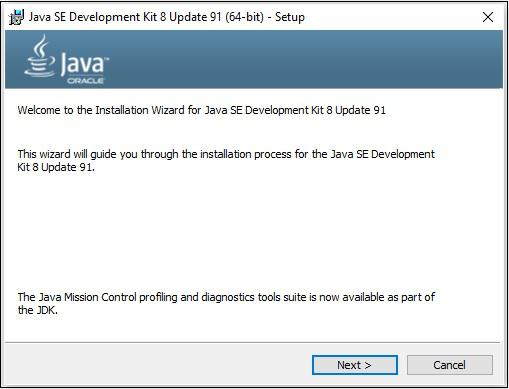
Accept the license agreement and download the required software by clicking on its respective link.

For example, if you are working on a windows 64-bit Operating System then you need to download the JDK version highlighted in the following screenshot.

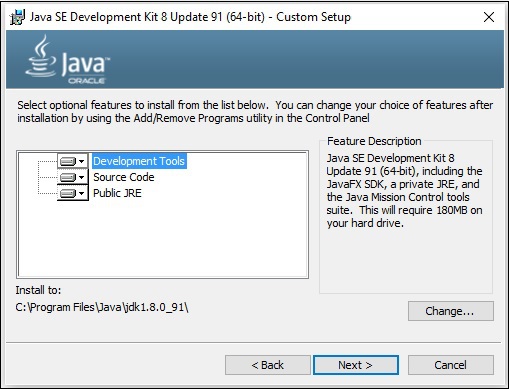


On clicking the highlighted link, the Java8 Development Kit suitable for Windows 64-bit Operating System will be downloaded onto your system.

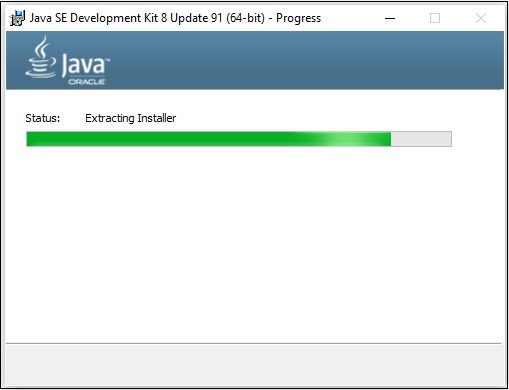
**Step 3** − Run the downloaded binary executable file to start the installation of JDK8.



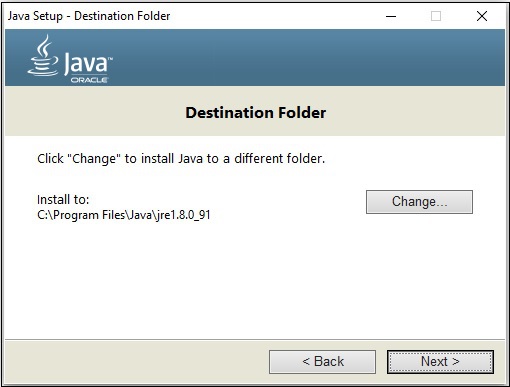
**Step 4** − Choose the Installation Directory.



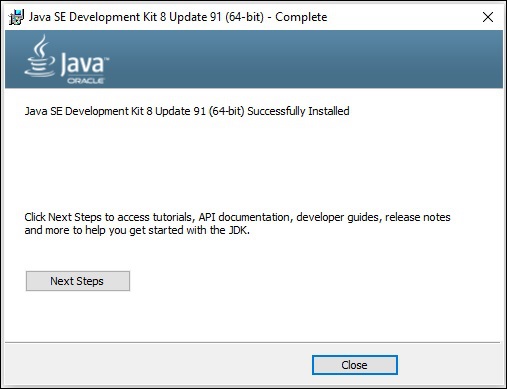
**Step 5** − On selecting the destination folder and clicking Next, the JavaFX installation process starts displaying the progress bar as shown in the following screenshot.



**Step 6** − Change the installation directory if needed, else keep the default ones and proceed further.



**Step 7** − Finish the installation process by clicking the Close button as shown in the following screenshot.

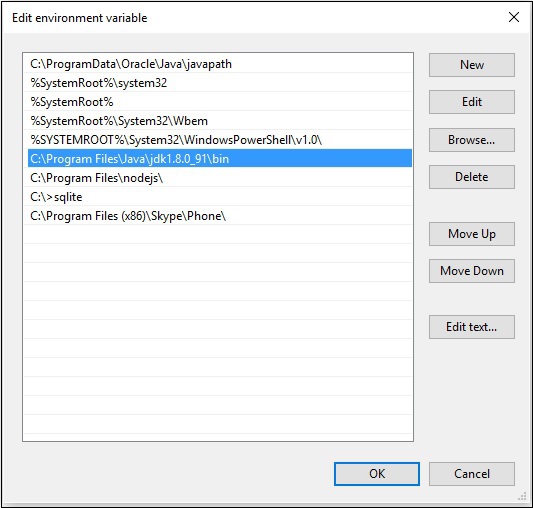


Setting up the Path for Windows

After installing Java, you need to set the path variables. Assume that you have installed Java in **C:\Program Files\java\jdk1.8.0\_91**directory.

Now you can follow the steps that are given below −

* Right-click on 'My Computer' and select 'Properties'.
* Click on the 'Environment Variables' button under the 'Advanced' tab.
* Now, alter the 'Path' variable so that it also contains the path to the Java executable. For Example, if the path is currently set to 'C:\WINDOWS\SYSTEM32', then change your path to read 'C:\WINDOWS\SYSTEM32; C:\Program Files\java\ jdk1.8.0\_91\bin'.



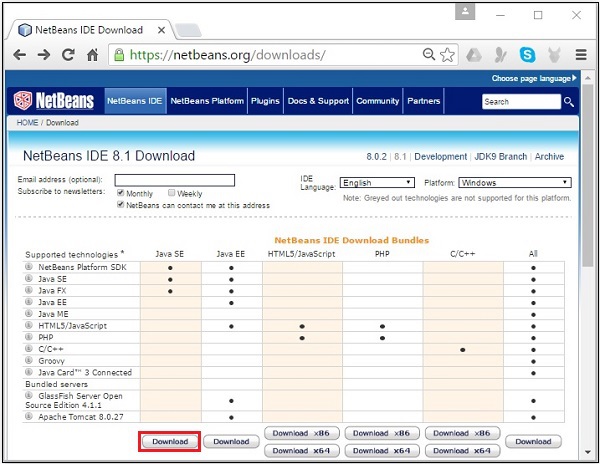
Setting NetBeans Environment of JavaFX

**NetBeans8** provides inbuilt support for JavaFX. On installing this, you can create a JavaFX application without any additional plugins or JAR files. To set up the NetBeans environment, you will need to follow the steps that are given below.

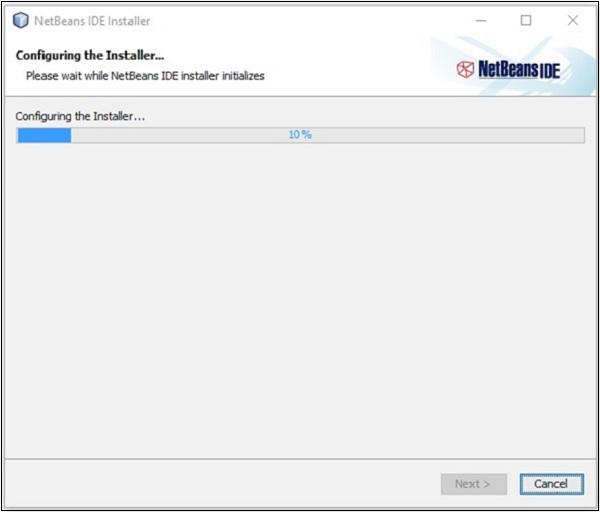
**Step 1** − Visit the [NetBeans website](https://netbeans.org/) NetBeans website and click the Download button in order to download the NetBeans software.



**Step 2** − On clicking **Download**, you will get to the Downloads page of the NetBeans software, which provides NetBeans bundles for various Java applications. Download the NetBeans software for **JavaSE** as shown in the following screenshot.

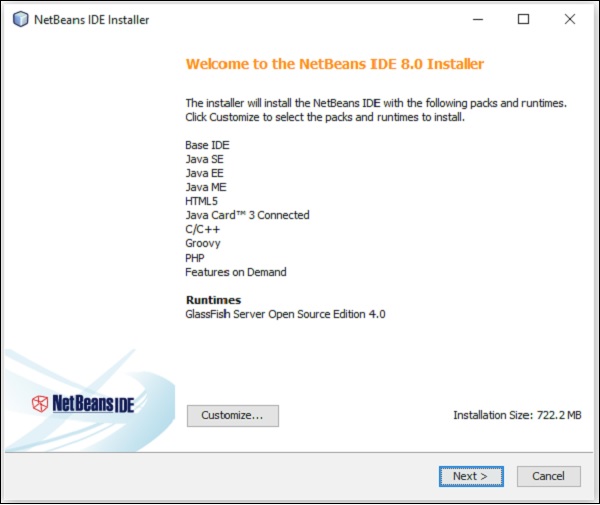


**Step 3** − On clicking this button, a file named **netbeans-8.0-windows.exe**will be downloaded onto your system. Run this file in order to install it. On running this file, a NetBeans installer will start as shown in the following screenshot.

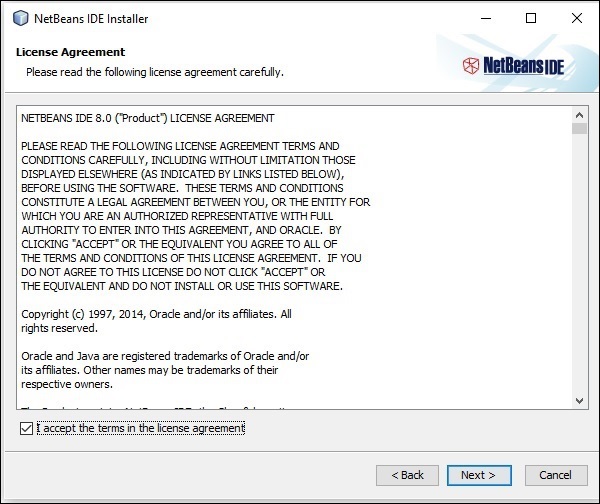


After completion of the configuration, you will see the **Welcome Page of the installer**.

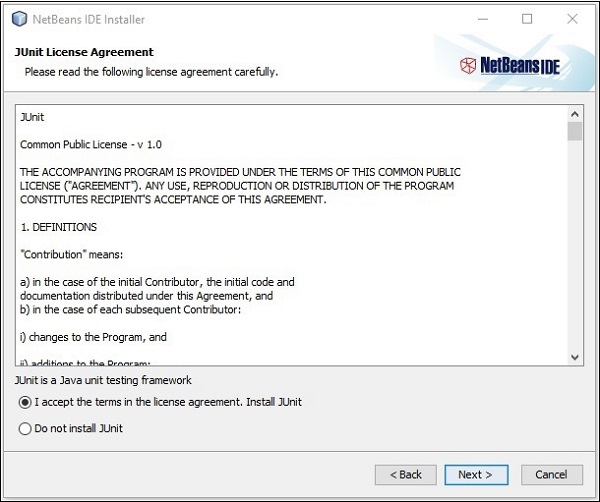
**Step 4** − Click the Next button and proceed with the installation.



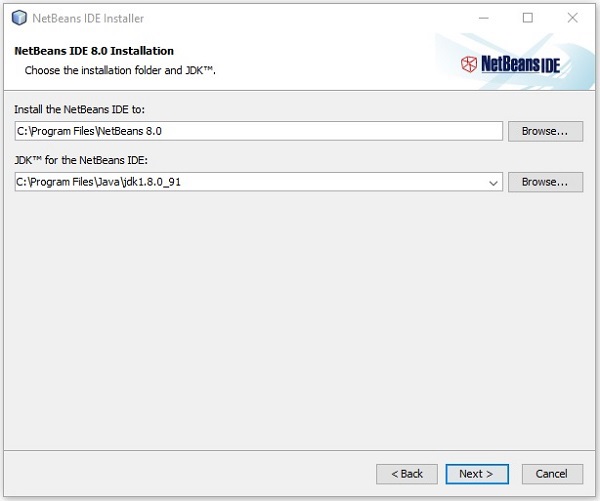
**Step 5** − The next window holds the **NETBEANS IDE 8.0 license agreement**. Read it carefully and accept the agreement by checking the checkbox at “I accept the terms in the license agreement” and then click the **Next** button.



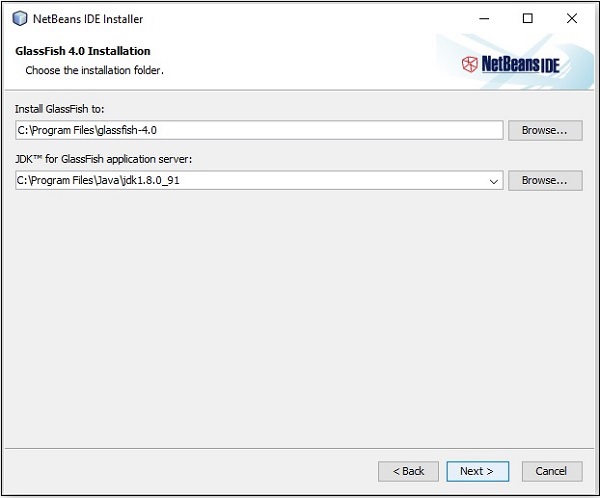
**Step 6** − In the next window, you will encounter the license agreement for **Junit**, accept it by selecting the radio button at “I accept the terms in the license agreement, Install JUnit” and click on **Next**.



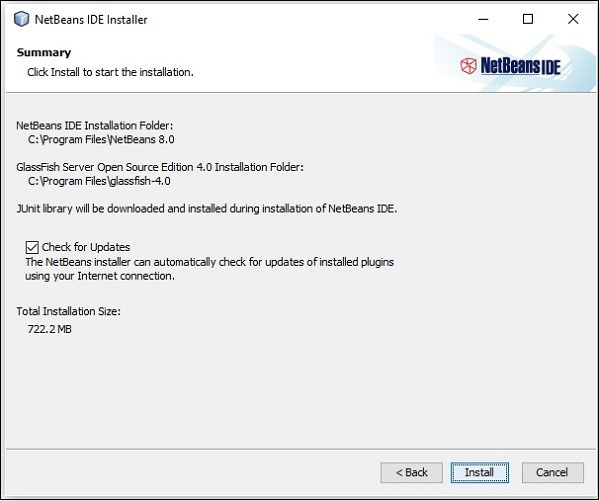
**Step 7** − Choose the destination directory where you need the Netbeans 8.0 to be installed. Furthermore, you can also browse through the directory where **Java Development Kit** is installed in your system and click on the **Next**button.



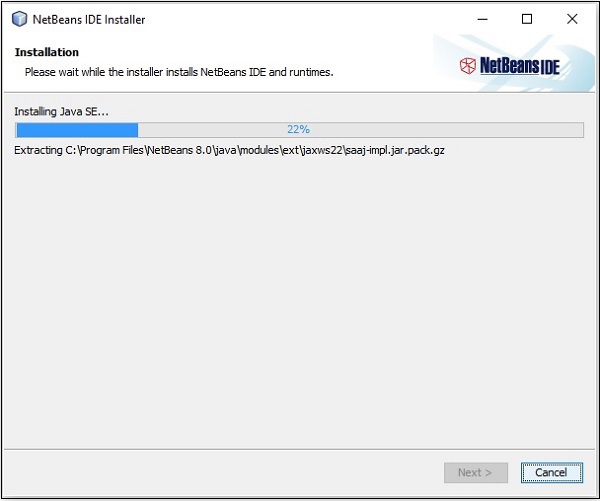
**Step 8** − Similarly, choose the destination directory for **Glassfish Server**installation. Browse through the Java Development Kit directory (now for Glassfish Reference) and then click **Next**.



**Step 9** − Check the **Check for Updates** box for automatic updates and click the Install button to start the installation.

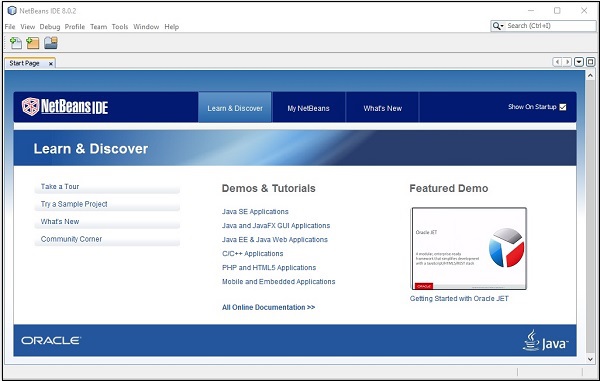


**Step 10** − This step starts the installation of NetBeans IDE 8.0 and it may take a while.

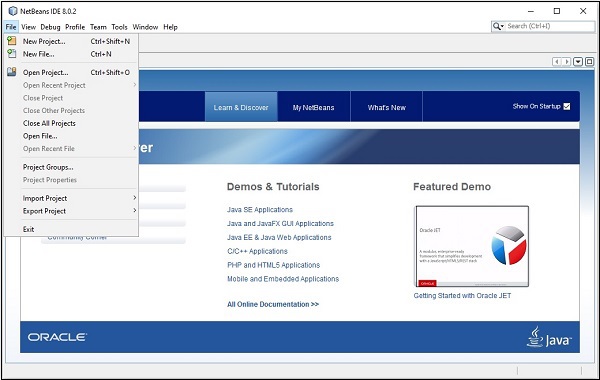


**Step 11** − Once the process is complete, click the **Finish** button to finish the installation.

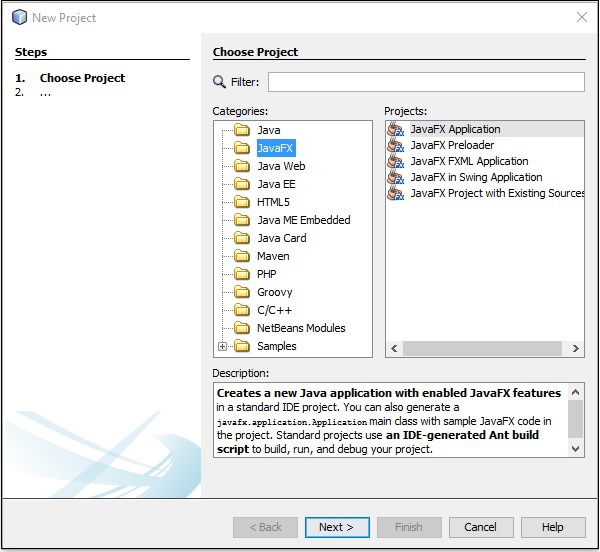
**Step 12** − Once you launch the NetBeans IDE, you will see the start page as shown in the following screenshot.



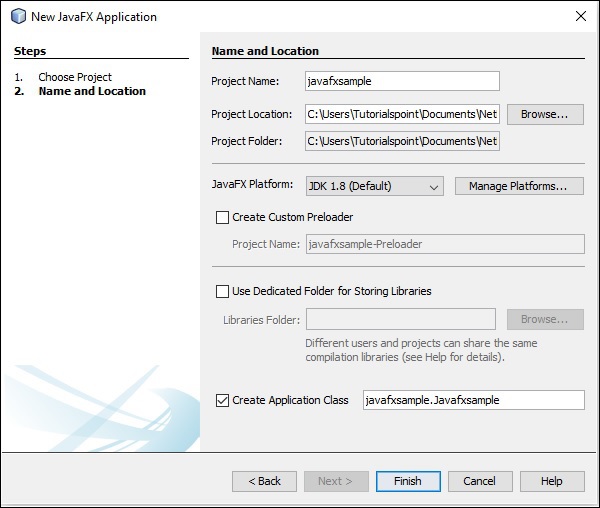
**Step 13** − In the file menu, select **New Project**… to open the New project wizard as shown in the following screenshot.



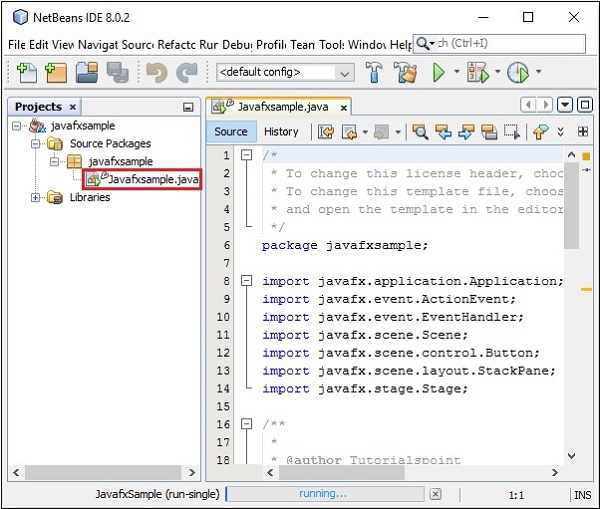
**Step 14** − In the **New Project** wizard, select **JavaFX** and click on **Next**. It starts creating a new JavaFX Application for you.



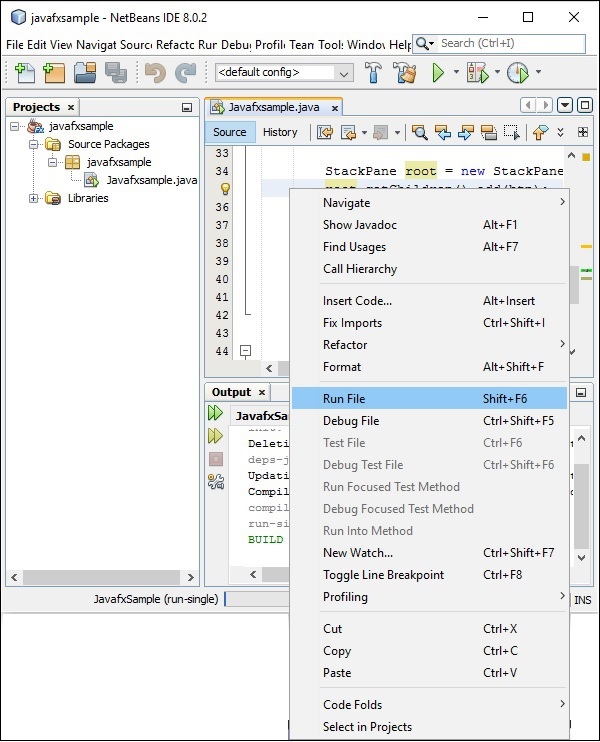
**Step 15** − Select the name of the project and location of the project in the **NewJavaFX Application** window and then click **Finish**. It creates a sample application with the given name.



In this instance, an application with a name **javafxsample** is created. Within this application, the NetBeans IDE will generate a Java program with the name **Javafxsample.java**. As shown in the following screenshot, this program will be created inside NetBeans Source Packages **→ javafxsample**.



**Step 16** − Right-click on the file and select **Run File** to run this code as shown in the following screenshot.

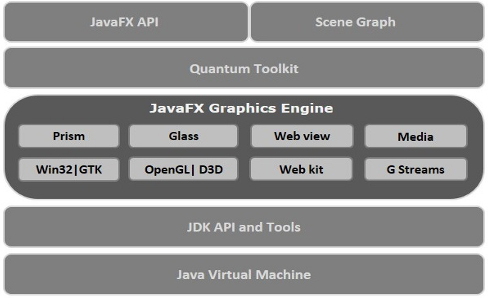


This automatically created program contains the code which generates a simple JavaFX window having a button with the label **Say ‘Hello World’** in it. Every time you click on this button, the string **Hello World** will be displayed on the console as shown below.



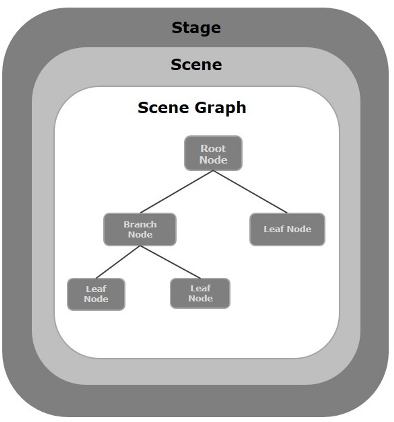
JavaFX Architecture:

Architecture of JavaFX API is shown below. Here you can see the components that support JavaFX API.



JavaFX Application Structure

In general, a JavaFX application will have three major components namely **Stage, Scene** and **Nodes** as shown in the following diagram.



Stage

A stage (a window) contains all the objects of a JavaFX application. It is represented by **Stage** class of the package **javafx.stage**. The primary stage is created by the platform itself. The created stage object is passed as an argument to the **start()** method of the **Application** class (explained in the next section).

A stage has two parameters determining its position namely **Width** and **Height**. It is divided as Content Area and Decorations (Title Bar and Borders).

There are five types of stages available −

* Decorated
* Undecorated
* Transparent
* Unified
* Utility

You have to call the **show()** method to display the contents of a stage.

Scene

A scene represents the physical contents of a JavaFX application. It contains all the contents of a scene graph. The class **Scene** of the package **javafx.scene** represents the scene object. At an instance, the scene object is added to only one stage.

You can create a scene by instantiating the Scene Class. You can opt for the size of the scene by passing its dimensions (height and width) along with the **root node** to its constructor.

Scene Graph and Nodes

A **scene graph** is a tree-like data structure (hierarchical) representing the contents of a scene. In contrast, a **node** is a visual/graphical object of a scene graph.

A node may include −

* Geometrical (Graphical) objects (2D and 3D) such as − Circle, Rectangle, Polygon, etc.
* UI Controls such as − Button, Checkbox, Choice Box, Text Area, etc.
* Containers (Layout Panes) such as Border Pane, Grid Pane, Flow Pane, etc.
* Media elements such as Audio, Video and Image Objects.

The **Node** Class of the package **javafx.scene** represents a node in JavaFX, this class is the super class of all the nodes.

As discussed earlier a node is of three types −

* **Root Node** − The first Scene Graph is known as the Root node.
* **Branch Node/Parent Node** − The node with child nodes are known as branch/parent nodes. The abstract class named **Parent** of the package **javafx.scene** is the base class of all the parent nodes, and those parent nodes will be of the following types −
  + **Group** − A group node is a collective node that contains a list of children nodes. Whenever the group node is rendered, all its child nodes are rendered in order. Any transformation, effect state applied on the group will be applied to all the child nodes.
  + **Region** − It is the base class of all the JavaFX Node based UI Controls, such as Chart, Pane and Control.
  + **WebView** − This node manages the web engine and displays its contents.
* **Leaf Node** − The node without child nodes is known as the leaf node. For example, Rectangle, Ellipse, Box, ImageView, MediaView are examples of leaf nodes.

It is mandatory to pass the root node to the scene graph. If the Group is passed as root, all the nodes will be clipped to the scene and any alteration in the size of the scene will not affect the layout of the scene.

## MySQL Database

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## Administrative MySQL Command

Here is the list of the important MySQL commands, which you will use time to time to work with MySQL database −

* **USE Databasename** − This will be used to select a database in the MySQL workarea.
* **SHOW DATABASES** − Lists out the databases that are accessible by the MySQL DBMS.
* **SHOW TABLES** − Shows the tables in the database once a database has been selected with the use command.
* **SHOW COLUMNS FROM *tablename:*** Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.
* **SHOW INDEX FROM tablename** − Presents the details of all indexes on the table, including the PRIMARY KEY.
* **SHOW TABLE STATUS LIKE tablename\G** − Reports details of the MySQL DBMS performance and statistics.

Important MySQL Commands :

## Commands

* Access monitor: mysql -u [username] -p; (will prompt for password)
* Show all databases: show databases;
* Access database: mysql -u [username] -p [database] (will prompt for password)
* Create new database: create database [database];
* Select database: use [database];
* Determine what database is in use: select database();
* Show all tables: show tables;
* Show table structure: describe [table];
* List all indexes on a table: show index from [table];
* Create new table with columns: CREATE TABLE [table] ([column] VARCHAR(120), [another-column] DATETIME);
* Adding a column: ALTER TABLE [table] ADD COLUMN [column] VARCHAR(120);
* Adding a column with an unique, auto-incrementing ID: ALTER TABLE [table] ADD COLUMN [column] int NOT NULL AUTO\_INCREMENT PRIMARY KEY;
* Inserting a record: INSERT INTO [table] ([column], [column]) VALUES ('[value]', [value]');
* MySQL function for datetime input: NOW()
* Selecting records: SELECT \* FROM [table];
* Explain records: EXPLAIN SELECT \* FROM [table];
* Selecting parts of records: SELECT [column], [another-column] FROM [table];
* Counting records: SELECT COUNT([column]) FROM [table];
* Counting and selecting grouped records: SELECT \*, (SELECT COUNT([column]) FROM [table]) AS count FROM [table] GROUP BY [column];
* Selecting specific records: SELECT \* FROM [table] WHERE [column] = [value]; (Selectors: <, >, !=; combine multiple selectors with AND, OR)
* Select records containing [value]: SELECT \* FROM [table] WHERE [column] LIKE '%[value]%';
* Select records starting with [value]: SELECT \* FROM [table] WHERE [column] LIKE '[value]%';
* Select records starting with val and ending with ue: SELECT \* FROM [table] WHERE [column] LIKE '[val\_ue]';
* Select a range: SELECT \* FROM [table] WHERE [column] BETWEEN [value1] and [value2];
* Select with custom order and only limit: SELECT \* FROM [table] WHERE [column] ORDER BY [column] ASC LIMIT [value];(Order: DESC, ASC)
* Updating records: UPDATE [table] SET [column] = '[updated-value]' WHERE [column] = [value];
* Deleting records: DELETE FROM [table] WHERE [column] = [value];
* Delete *all records* from a table (without dropping the table itself): DELETE FROM [table]; (This also resets the incrementing counter for auto generated columns like an id column.)
* Delete all records in a table: truncate table [table];
* Removing table columns: ALTER TABLE [table] DROP COLUMN [column];
* Deleting tables: DROP TABLE [table];
* Deleting databases: DROP DATABASE [database];
* Custom column output names: SELECT [column] AS [custom-column] FROM [table];
* Export a database dump (more info [here](http://stackoverflow.com/a/21091197/1815847)): mysqldump -u [username] -p [database] > db\_backup.sql
* Use --lock-tables=false option for locked tables (more info [here](http://stackoverflow.com/a/104628/1815847)).
* Import a database dump (more info [here](http://stackoverflow.com/a/21091197/1815847)): mysql -u [username] -p -h localhost [database] < db\_backup.sql
* Logout: exit;