

# Java Jazz up

A B E T T E R   W A Y   T O   L E A R N   P R O G R A M M I N G

**Introduction to JavaFX Technology**

**Releases of JavaFX**

**Features and Benefits**

**JavaFX Scripts**

**JavaFX Mobile**

**JavaFX API**

**Examples**



# JavaFX

# India's Cheapest web Service Provider



## OUR SERVICES

- ERP Soluiotns
- Software Solutions
- Web Development
- Web Designing
- Web Redesigning
- Domain Registration
- Web Promotion
- SEO
- Article Writing
- Blog Writing
- News Writing
- SEO Copywriting
- Technical Documentation
- E-Commerce Solutions
- CRM
- Outsourcing

Extend your reach  
with our solutions...

**"Optimism with determination lets you  
hit the goal harder"**

**Published by**

**RoseIndia**

**JavaJazzUp Team**

**Editor-in-Chief**

Deepak Kumar

**Sr. Editor-Technical**

Ravi Kant

**Graphics Designer**

Santosh Kumar

**Register with JavaJazzUp  
and grab your monthly issue**

**"Free"**

## Editorial

Dear Readers,

Welcome back to the Aug 2008 issue of the Java Jazz Up. We are heartily thankful to all our readers for giving so much positive response. For enhancing and updating your knowledge, we are bringing 'JavaFX', the Rich Internet Application, development technology in this August 2008 edition. This special issue of this Java Magazine is intended to provide readers knowledge on almost every aspect of JavaFX Technology. In this issue, we have described introduction to JavaFX, Features and benefits, JavaFX products (JavaFX Scripts, JavaFX Mobile), Comparison to the other existing technologies from different providers, System requirements, Software needed, Creating User Interfaces in JavaFX, Working with JavaFX Classes and Objects, Developing "Hello World Example" and many example applications. We hope that this new edition will provide you adequate knowledge of JavaFX. JavaFX is being extremely popular as Rich Internet Application development technology in the present development world.

Like earlier editions, we have categorized the whole topics in sections and each section is decorated with different colors and images that would certainly lure readers while reading technological stuffs. This will create interest in readers and users for learning it easily. We are also providing our online Java Jazz Up magazine in PDF format that you can view and even download it as a whole or a part. This PDF version provides you a different experience. Please send us your valuable feedback about this issue and participate in the reader's forum with your problems and issues concerned with the topics you want us to include in our next issues.

**Editor  
Deepak Kumar  
Java Jazz up**

# Content

- 05    Introduction to JavaFX Technology |** JavaFX is the latest innovation in the World of Java-based technology. This special issue of this Java Magazine is intended to provide readers knowledge on almost every aspect of JavaFX Technology.
- 07    Releases of JavaFX|** On July 31, 2008 the Sun officials announced the availability of the JavaFX Preview Release. This JavaFX Preview Release introduce web designers and Java developers with a preview of the runtime and tools with the help of which they can develop and deploy Rich Internet Applications (RIA) on the Java platform.
- 09    Features and Benefits |**JavaFX provides the web designers a high declarative scripting language called JavaFX Script, for creating and developing rich Internet applications basically for desktop browsers and mobile devices.
- 10    JavaFX Products (Implementation) |** JavaFX is a technology for developing Rich Internet Content Application basically targeted to the Web Developers and Designers.
- 11    JavaFX Scripts|** The new JavaFX Scripting language invented by Sun Microsystems is an output of its F3 Project. With the introduction of this special declarative language Sun actually announces its footstep towards entering the RIA market with a definitive goal in making user interface development scenario to be featured by powerful visual impact in graphical context.
- 12    JavaFX Mobile |** JavaFX mobile is invented by Sun as an operating system based on Java technologies and Open Source Linux Platform.
- 13    Competition between adobe, Microsoft and Sun Microsystem |** Sun Microsystems has invented the JavaFX technology to emerge into the field of creating Rich Internet Application. Java can be a formidable player in this regard with this family of product which covers Java Development from the desktop to the web to mobile devices.
- 15    System Requirements|** We are using "**NetBeans IDE 6.1 with JavaFX**" and **Java SE 6 Development Kit (JDK)** to develop the examples.
- 16    Software Needed|** Once you have downloaded the required software, the next step is to install it on your development box. First install JDK and configure it on the development box.
- 17    Introduction to JavaFX API |** JavaFX provides common APIs and device specific APIs. JavaFX common APIs provides common functionalities across all the devices and device specific APIs are helpful to provide device specific functionalities.
- 19    Creating User Interfaces in JavaFX |** JavaFX SDK Runtime provides libraries and classes for JavaFX platform to create rich application interfaces. Developers, designers and GUI architects can create user interfaces easily for Rich Internet Applications.
- 30    Developing "Hello World Example" in JavaFX |** There are the step-by-step procedure to create first "Hello World" example.
- 37    Advanced Examples to JavaFX |** After developing Hello application, we are now going to create some more examples having different features.
- 59    Advertise with Us |** We are the top most providers of technology stuffs to the java community. Our technology portal network is providing standard tutorials, articles, news and reviews on the Java technologies to the industrial technocrats.
- 60    Valued JavaJazzup Readers Community |** We invite you to post Java-technology oriented stuff. It would be our pleasure to give space to your posts in JavaJazzup.

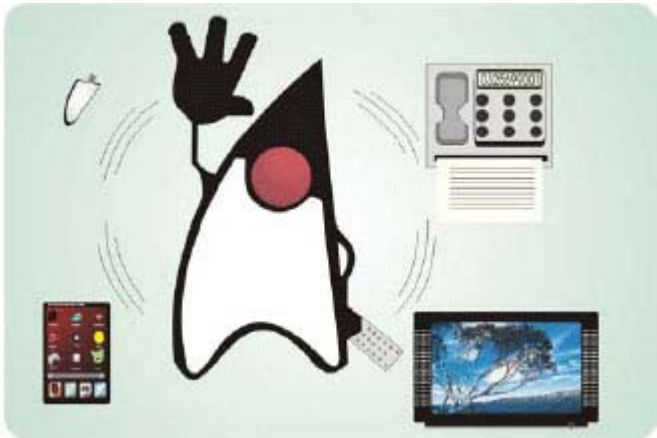


# Introduction to JavaFX Technology

JavaFX is the latest innovation in the World of Java-based technology. This special issue of this Java Magazine is intended to provide readers knowledge on almost every aspect of JavaFX Technology.

JavaFX is a family of products developed by Sun Microsystems for creating Rich Internet Applications(RIA). It is intended to enhance the content authors. With the use of this technology web designers can develop of rich interactive applications with more compelling media contents than ever before. JavaFX is to be used for desktop and mobile devices, TV and for other related platforms too. It is believed to bring accuracy in RIA designs incorporating a wide range of devices and screens. JavaFX technology is targeted towards a large number of mass basically non-programmers or users for developing or using their applications.

This latest java framework was released by Sun Microsystems in the Spring 2007. It mainly comprises of two components: JavaFX Scripting Language and JavaFX Mobile. The general product release is yet to come into the market. JavaFX script is a declarative scripting language intended to give web developers a boost to develop user interfaces with 2D & 3D graphics animations, audio & video. It has a high degree of interactivity with Java Classes.



The second component of JavaFX framework is JavaFX Mobile. It is a complete software solution for mobile devices. JavaFX Mobile is an Java based operating system for mobile devices. It was originally developed by SavaJe Technologies. Sun Microsystems purchased it from Savaje technologies in April 2007. JavaFX Mobile is featured by a Java SE (Standard Edition) and Java ME (Mobile Edition) implementation running on top of a Linux kernel.

## JavaFX Platform:

The JavaFX platform is a runtime dealing with a wide range of devices intended to enhance rich client capability. The JavaFX runtimes include-

- JavaFX Desktop- This is for the use in desktop browsers and to develop desktop applications. It is said to be available in the Fall 2008.
- JavaFX mobile- It will be available by Spring 2009. It is meant for the mobile devices that runs on the Java-based technology.
- JavaFX TV- It is for the application running on the TV platform. It will be made available to the consumers by early 2010.

# Introduction to JavaFX Technology

## JavaFX Tools:

JavaFX tools encourage the Developers and Designers to take freedom in creating interactive content and applications for the users of Rich Internet Applications incorporating with a wide range of devices. JavaFX tools are-

- JavaFX development tools. This include plug-ins for IDE (Integrated development environment)s like NetBeans, Eclipse etc.
- Project Nile. This will allow to export graphical assets from professional design tools like Adobe Flash directly into JavaFX. **Project Nile** includes plugins for Adobe Photoshop and adobe Illustrator for faster delivery of visual elements in to JavaFX applications, and make web developers work much easier.
- JavaFX designer Authoring tool will be available for Visual Designers.



# Releases of JavaFX

On July 31, 2008 the Sun officials announced the availability of the JavaFX Preview Release. This JavaFX Preview Release introduce web designers and Java developers with a preview of the runtime and tools with the help of which they can develop and deploy Rich Internet Applications (RIA) on the Java platform. It is available at <http://www.javaafx.com>. The web developers can download the JavaFX Preview release, they can develop their applications using JavaFX scripts to have an insight into the new technology and deliver their feedback.

## JavaFX SDK:

The JavaFX Software Development Kit (SDK) is basically meant for the web designers and developers. They can create rich internet applications for desktop and browsers using it. A Preview Version of the JavaFX SDK is already available for download. The components of JavaFX SDK are-

- JavaFX Preview SDK- It includes the JavaFX compiler and runtime tools, with 2D graphics and media libraries.
- Netbeans IDE 6.1 with JavaFX- It provides a mean to the developers to create and debug JavaFX applications.
- Project Nile: It includes a set of Plugins for Adobe Photoshop and Adobe Illustrator from which the graphical features can be exported to JavaFX applications.
- Java Runtime Environment 6 Update 10 Beta Release- This is a version of Java Runtime Environment (JRE). It includes a Java browser plug-in that with the help of which users can drag and drop JavaFX applications from the browser to desktop.

## The JavaFX Preview SDK is free and available under a 60 day evaluation license.

This release is only an alpha release as said by the Sun officials in the JavaOne conference. This is basically meant for individual use to help the users have a clear review of the new technology. It is not meant for development of commercial applications at time being. Beacause updatation and changes are to be made in the later version according to the internal development in the company and as based on the customer feedback.

The Sun executives are planning to make the JavaFX open source which is in demand in today's world of technology to make it more comprehensive and bring continuous development to the features of the technology. The features of JavaFX is meant for the enhancemant of the use of Java on small devices which was one of the main objectives at the time of the creation of the language.

## OpenJavaFX Compiler Source Code released by Sun Microsystems:

Though from the time of announcement of JavaFX, Sun Microsystems has been quiet about the new technology, recently the company has released OpenJavaFX compiler. OpenJavaFX compiler source code is now available along with online demonstrations. JavaFX is intended to enhance the GUI Applications based on the Swing GUI toolkit for Java. The Beta Release of JavaFX is intended for release in the next JavaOne Conference. The now available code is only the early access code for individual developers which is called as Sub-Alpha code, as the Sun officials reported.

The JavaFX compiler converts the JavaFX script code into compiled java code. One has to compile the entire JavaFX program, thus by getting all the code in Java which makes the compilation faster. The added NetBeans Plug-ins will enable JavaFX to be developed in the Interface Development

## Releases of JavaFX

Environment (IDE). According to **Chet Haase, Sun Java Chief Client Architect**, JavaFX programs will be available as NetBeans Projects. Both the NetBeans 5.5 and 6.0 plug-ins are available now. The NetBeans 6.0 Version has features of JavaFXPad integration tool which enables the web developers in creating graphical components using JavaFX Script.

JavaFX is intended to make the development of custom-based applications more faster and easier. As Chet Haase said, "It will include websites with rich animations and user interfaces that are more dynamic than a typical forms-based application".

JavaFX is likely to extend its capability beyond the advantages offered by AJAX. AJAX is not rich in user interface development as it is still now more about creating static web pages. But JavaFX is intended to more dynamic user interface development.

Sun's business strategy is to make the further development of the JavaFX Compiler in the open source community. The current JavaFX Preview SDK libraries will remain on OpenJFX.com. The efforts subject to the enhancement of the JavaFX platform have to be made by Sun with its internal development and in the further releases new features are to be consequently added.



# Features and Benefits

## Features of JavaFX Technology:

- JavaFX provides the web designers a high declarative scripting language called JavaFX Script, for creating and developing rich Internet applications basically for desktop browsers and mobile devices. It can be applied for TV and other consumer platforms also.
- JavaFX Script is created to have the feature of integrated development with other Java application components which run on both client and server platforms. This will enable the users and developers of Java to experience a rich end to end interaction.
- JavaFX brings about the security advantages to its customers. As they can access any kind of documents like a picture file, a music file or a word document on their desktop securely.
- JavaFX technology offers the developers the opportunity to work altogether with the web designers with a combined effort to create more enhancing and richer internet application thus by bringing the presentation layer come into play.

## Advantages:

- JavaFX script is based on Java's best features. Platform indepenency is one of Java's widely known traits. Once JavaFX scripts are written they can be run on any platform thus making it portable beacause of which Java has become so much popular today. JavaFX is also intended to provide application security and ubiquitous distribution. JavaFX has its ability lying in leveraging the power of Java Platform.
- JavaFX Script's one of the advantages is that with the help of it the web designers and developers can create rich content applications at a faster rate than ever and that too in an easier mode. These applications can be used for desktops, browsers, mobile devices, set-top boxes etc. and in future for other platforms too.
- By using JavaFX Script the web developers can create now more compelling and dynamic application and interfaces which are rich in visual graphical contents. It also binds the capabilty of other Java components to give designers and developers more flexibility in making applications attractive to the users.

# JavaFX Produits (Implementation)

JavaFX is a technology for developing Rich Internet Content Application basically targeted to the Web Developers and Designers. Sun Microsystems has released two components of JavaFX, viz., JavaFX Scripts and JavaFX Mobile at the recent JavaOne Conference. This is only an alpha release and the Beta release is said to come into existence in the Fall 2008 whether the date for the General Final Release of the produits is yet to be announced.

## **JavaFX Scripts:**

JavaFX Script is said to be more productive and easier than Java. But it has a close integration with the other components of Java too which will enable the developers to bring the best features of Java come into play. The applications created in JavaFX Script runs on devices powered by Java.

Sun's executive vice-president of software Rich Green said that the product recently announced is for the use of consumers and individuals but not for the enterprises. He also added that the power of JavaFX script to build rich content application will bring more number of people in the scenario being attracted towards the Java Platform.

JavaFX is basically created to assist the web designers and content authors in developing user interfaces rich in graphics. This simple and easy-to-learn declarative scripting language allows the content authors to build the applications in a format which looks like the actual layout of the graphical user interfaces. To create complex and more powerful applications easily and quickly the experienced and expert programmers can find this technology a handful one.

## **JavaFX Mobile:**

In today's world, the mobile technology is fast growing and delivering good quality services to consumers by bringing more and more facilities to be incorporated with. Recent rapid technological development has brought the features of accessed rich internet application to the mobile users. With the use of such applications consumers can download the internet contents to their desktops and mobiles. Sun Microsystems has stepped into this space with aim to bring enhancements to the developers of such applications to allow them to create useful and dynamic graphical user interfaces across a wide range of mobile devices. JavaFX mobile is based on the Java ME platform to empower the developers with the facilities in context with the market driven strategies.

JavaFX mobile was originally designed and developed by SavaJe technologies. It was purchased by later by Sun in April 2007. It is a powerful technology assisted by Java platform and flexible across a wide variety of mobile devices. JavaFX mobile applications run on all mobile devices powered by Java. JavaFX is a kind of operating system featuring Java ME implementation which runs on the top of a Linux Kernel. It is providing a platform for devices like smartphones and PDAs. Via OEM (Original Equipement Manufaturer) licenses provided by the Sun Microsystems the JavaFX mobile is available to the enterprises.

# JavaFX Scripts

The new JavaFX Scripting language invented by Sun Microsystems is an output of its F3 Project. With the introduction of this special declarative language Sun actually announces its footstep towards entering the RIA market with a definitive goal in making user interface development scenario to be featured by powerful visual impact in graphical context. They are targetting the desktop and mobile markets across the globe. The new language is said to be more easier to learn basically with a view to help the web designers in creating their applications. JavaFX Script is intended to make the use of Java's basic features along with the binding of the other components of Java. JavaFX also provides the Plugins like NetBeans and Eclipse to work with integrated development environment(IDE)s. It is released under the GNU General Public License(GPL), via OpenJFX project sponsored by Sun.

## Features of JavaFX Script:

- In developing a GUI, a developer can use a declarative syntax when writing in JavaFX for specifying GUI components to match with the the actual layout of the GUI.
- JavaFX script provides automatic databinding and incremental evaluation. Developers can have the feature of autmatic synchronization of the application data and GUI components so as to create and configure individual components more easily.
- JavaFX Script eanbles the web developers to work in IDEs, e.g., NetBeans and Eclipse, by enhancing the implementation of Java development.
- JavaFX Script is a statically typed language. It brings the features of Java, viz, reuse of codes and encaptulation. Programmers will be able to create and maintain large programs using JavaFX.
- JavaFX Script is able to support complex GUIs of any size.
- JavaFX enables the enhancement of Swing which a GUI toolkit.

# JavaFX Mobile

JavaFX mobile is invented by Sun as an operating system based on Java technologies and Open Source Linux Platform. It is intended to provide mobile users the best features of today's multimedia applications. JavaFX Mobile Technology will enable the manufacturers of mobile devices to create rich internet applications in context of mainly video and multimedia. Now users of such applications can enjoy the freedom of downloading the rich internet contents on all type of devices like desktop browsers, mobile handsets and even TV and other devices in future. Java brings the best use of its technological advantages and features thus by making the development of Graphical User Interfaces easier and faster than ever over almost all the platforms. JavaFX mobile applications will run ideally on all sort of platform and devices based and built on the Java technology. JavaFX Mobile is said to be available in the market in the next year by Sun officials. The distribution of JavaFX mobile will be delivered by selling out OEM licences to the manufacturers.

However, JavaFX mobile was first introduced by SavaJe Technologies. Sun purchased it from SavaJe in April 2007. Although it has been said that JavaFX mobile applications can be run on all platforms still cloud of uncertainties is floating over the scenario as the demonstration of the technology at the recent JavaOne conference held showed by the Sun officials revealed that it requires a predefined multitasking kernel underneath. The demonstration was carried out on devices like Nokia N800 Internet Tablet and SavaJe Jasper phone based on Linux Platform. Therefore, a thin line of doubt still hitting the mind of the experts if this JavaFX applications will be supported by all kind of embedded platforms or it will run only on top of a Linux kernel.

But according to the official report apart from some low-level services this Linux based application will be supported by all platforms powered by Java. Uses of JavaFX Script in building the applications will make the JavaFX mobile technology highly portable through all kind of devices incorporating all desktop browsers and web applications. The "Write Once, Run Anywhere" feature of Java is bringing all possible enhancements in the further open source development of the technology.



# Competition between adobe, Microsoft and Sun Microsystem

## **The Market Perspective- Competition among the Three Giants Adobe, Microsoft and Sun Microsystems:**

Sun Microsystems has invented the JavaFX technology to emerge into the field of creating Rich Internet Application. Java can be a formidable player in this regard with this family of product which covers Java Development from the desktop to the web to mobile devices. But Sun is not alone in the run. As two other giants Adobe and Microsoft are also entering into the business with their respective product releases into the market. Adobe has recently added the final preview to their release of Flex3 as Beta version 3. Simultaneously Microsoft has also taken a great step towards capturing the RIA market by announcing their Silverlight2 Beta2 release recently. Therefore, to stay in the long run Sun has to go through a tough competition with their fellow competitors.

Let us analyse why Sun has adopted a new language ahead of already existing strong programming technologies in context of multimedia development. This very question might have stressed many of our readers. The constant and fast growing RIA markets in today's world is concerning more about the web designers and not relying alone on the developers in context of creating multimedia based application. The invention and use of declarative scripting languages has brought the advantages thus by providing much easier and faster development of content rich application and still restoring the basic powerful features of programmatic language. For example, a language called MXML which is based on XML is used in the creation of User Interfaces in Flex developed by Adobe. MXML is enriched by the feature of strong data binding capability of the language and uses ActionScript for programming. In Silverlight developed by Microsoft, another XML based language called XAML is used for UI application development. Silverlight is based on the .NET framework which facilitates the use of C#. VB.NET and ASP.NET. Similarly, JavaFX uses JavaFX Script which is a declarative scripting language like MXML and XAML to create UI easily and quickly. But JavaFX differs from MXML and XAML in specifying the GUI components. The code in JavaFX Script intends to match the actual layout of the GUI. Moreover, JavaFX also has the capability to extend the power of Java by integrating all other Java Components within it. JavaFX makes the best use of Swing toolkit. JavaFX targets the Flash designers in developing rich internet application.

Sun's Executive Vice President of Software Rich Green said that while JavaFX is basically concentrating on the User Interface Design and animation development in contrast of most of the the scripting languages are primarily concerned in web page creation. Thus it is targetting at a large volume of mass interacting with applications rich in graphical content and multimedia. According to the acknowledgement of the Sun officials JavaFX is somewhat similar in the use of enhanced graphical capability of Silverlight. But Silverlight differs in that it is mainly focused on Video-Streaming. The JavaFX mobile component of JavaFX technology is all set to boost the development of mobile devices with the added functionality of Java ME.

Although JavaFX Mobile has already created an impression of being technologically promising still it will have to fight a lot to secure its position in the business against the large coalitions of other corporate houses emerging in the rapidly developing Open Source Linux Community.

With 5 billion programmers in Java SE platform and nearly 2 billion programmers working in ME Platform, Java claims a high volume distribution all over the world. As the JavaFX is said to heavily rely on Java based platform with runtime SE and ME outcomings may be predicted as satisfactory. But although for JavaFX mobile the users will have the built-in Java SE/ME runtime, for the users of desktop browsers Sun has to rethink in making strategic decision to mobilise the market. There exists a problem for desktop browsers to download and updates of Java while using JavaFX and in contrast Flash has an advantage in the context of the easier installation process and more security.



## Competition between adobe, Microsoft and Sun Microsystem

The most important factor in the success of any of the three technologies lies in the release of the final general product into the market prior to that of the other two. Flex, Silverlight and JavaFX is hitting the industry at the same time thus creating more competition among themselves. All the three Corporate houses Adobe, Microsoft and Sun Microsystems are trying their best bring their products earlier into the market. While the perspective of all of them is much similar, i.e, to provide designers, developers and users with more rich and powerful internet content creation application, their approaches in achieving the goal in terms of technology make all the things turn around.

**Although it is so early to tell which will have the advantage over the other two technologies.**

There are two main camps in the software development in today's world, viz, one using the Microsoft's .NET framework and the other using Sun Microsystem's Java-based technology. Most of the large organizations tend skewed more heavily toward Java and J2EE frameworks. Though large organizations also use .NET framework, it is more widely used in the small organizations only. In this regard, JavaFX may enjoy advantage over Silverlight. But more effectively, actual pressure revolves round the factor of delivering quality products in a reduced or accelerated timeframe. Working within a specified budget limit is also a definite key factor. This kind of pressure situations has led the organizations a fundamental shift in power. Now most of the power houses are making their product open source to enhance its quality and thus by securing the integrity of such products.

# System Requirements

We are using “**NetBeans IDE 6.1 with JavaFX**” and **Java SE 6 Development Kit (JDK)** to develop the examples. NetBean IDE 6.1 provides nice GUI to get started with the JavaFX technology. NetBean IDE 6.1 also provides good sample applications to have a feel of the JavaFX language.

Actually we need a latest computer with at least following configuration.

	Windows	Macintosh
Processor	Intel® Pentium® 4, Intel Centrino®, Intel Xeon®, or Intel Core™ Duo (or compatible)	Multicore Intel processor
Operating System	Microsoft® Windows® XP with Service Pack 2 or Windows Vista™ Home Premium, Business, Ultimate, or Enterprise (certified for 32-bit editions)	Mac OS X 10.5.2+ (Leopard)
Ram	512MB (1GB recommended)	512MB (1GB recommended)
Hard-disk space	900 MB of available space minimum (1.1 GB recommended)	800 MB available space minimum (1.1 GB recommended)
Java Runtime	Java 6 update 7	Java 6 update 5

# Software Needed

**We just need following software on the development.**

- Java SE 6 Development Kit (JDK)
- NetBeans IDE 6.1 with JavaFX

You can download NetBean for JavaFX from <http://javafx.netbeans.org/> and JDK from <http://www.java.sun.com>.

Once you have downloaded the required software, the next step is to install it on your development box. First install JDK and configure it on the development box. Then install NetBeans IDE.

# Introduction to JavaFX API

JavaFX provides common APIs and device specific APIs. JavaFX common APIs provides common functionalities across all the devices and device specific APIs are helpful to provide device specific functionalities.

## Common APIs:

If you need to create cross platform Rich Internet Applications then you can use JavaFX Common APIs. Currently JavaFX Common APIs provide support for 2D graphics, Animation, Text and Media but not for audio, video, networking, local storage etc. The support for these components will be added in the coming days.

## Device specific APIs:

If you want to develop device specific Rich Internet Applications (for example, applications for desktop, mobile and TV) then you can use device specific APIs in addition to JavaFX common APIs. The device specific APIs provide more suitable functionalities for the device. For example, for desktop application you can use desktop specific APIs to achieve desktop specific look and feel and graphics support.

## Some of the JavaFX packages are described below:

<code>javafx.application</code>	Provides support for the JFC/Swing component architecture, application lifecycle support methods and attributes and defines the possible styles for a Window etc. It contains classes like Applet, Application, Frame, Window, Dialog etc.
<code>javafx.animation</code>	Contains classes used for animation effects. Attributes of any object (visual and non-visual) can be animated using its classes.
<code>javafx.ext.swing</code>	Contains wrappers for Swing components like Frame, ComboBox and List and provides layout constants. It provides classes like: Layout, Component, Container, Button, CheckBox, MenuItem, ComboBox, GridPanel, Icon, Label, ScrollableComponent, List, ListItem, Menu, RadioButton, RadioButtonMenuItem, ScrollPane, SequentialCluster, Slider, SwingApplication, SwingWindow, SwingDialog, TextField, SwingFrame, ToggleGroup etc.
<code>javafx.input</code>	Provides support for key events and mouse events and provides set of key codes for KeyEvent objects.
<code>javafx.lang</code>	Used for DeferredTask (an action that is executed at a later time), defining the duration of time, various functions for manipulating sequences.
<code>javafx.scene</code>	Base class for scene graph nodes. Provides classes for alignment and orientation of the UI elements and operate with fonts.
<code>javafx.scene.image</code>	Provides classes representing graphical images.
<code>javafx.scene.paint</code>	Contains classes defining color and different color patterns for graphics operations.

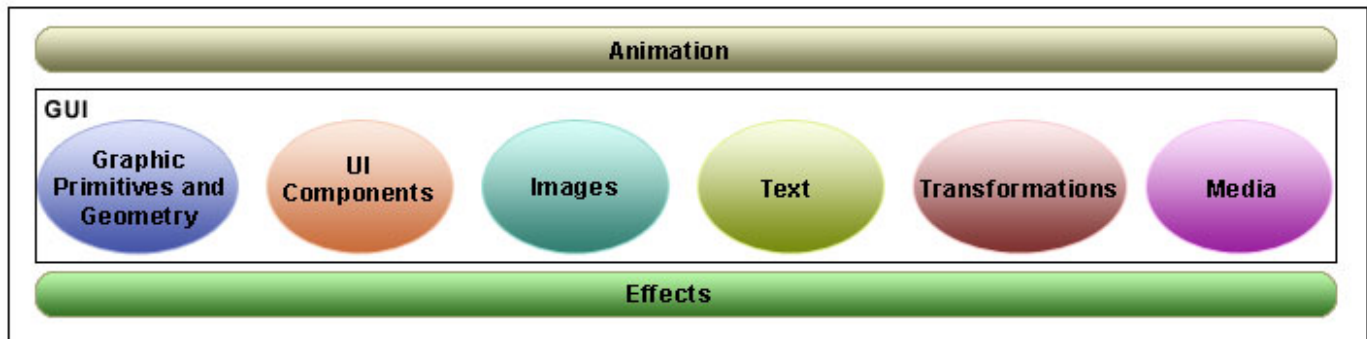
## Introduction to JavaFX API

<code>javafx.scene.transform</code>	Provides classes to apply transformations to objects. Provides functions to perform rotating, scaling, shearing, and translation for Affine objects.
<code>javafx.scene.layout</code>	Used for layout of GUI elements of the application.
<code>javafx.scene.geometry</code>	Provides classes for drawing operations like drawing lines, geometrical objects, curves, or shapes (Circle, Arc, Ellipse, Line, Polygon, Rectangle etc.).
<code>javafx.scene.text</code>	Provides classes to operate with text.
<code>javafx.scene.media</code>	Provides classes that are responsible for media playback.
<code>javafx.scene.effect</code>	Used for applying various effects to the application. Supports blurring, rendering shadows, color adjusting, blending, glowing, toning, inverting, reflecting, blooming, and flooding operations.
<code>javafx.scene.effect.light</code>	Provides classes for 2D and 3D lighting effects.
<code>javafx.util</code>	Provides utility class for localizing strings using JavaFX properties files.



# Creating User Interfaces in JavaFx

JavaFX SDK Runtime provides libraries and classes for JavaFX platform to create rich application interfaces. Developers, designers and GUI architects can create user interfaces easily for Rich Internet Applications. JavaFX provides rich text, 2D vector graphics, animation, audio, and video to help creating Rich Internet Applications. The structure of JavFX Runtime functionality is given below:



**JavaFx GUI Structure**

**Here is the list of JavaFX Runtime packages:**

- `javafx.animation`
- `javafx.ext.swing`
- `javafx.scene`
- `javafx.scene.effect`
- `javafx.scene.effect.light`
- `javafx.scene.geometry`
- `javafx.scene.image`
- `javafx.scene.layout`
- `javafx.scene.media`
- `javafx.scene.paint`
- `javafx.scene.text`
- `javafx.scene.transform`

# Working with JavaFX Classes and Objects

Here we will have a look over some programming facts before going to develop our JavaFX script applications. This will help you understand how the JavaFX script programming works.

## i) Scripts:

In JavaFX script programming language, a script doesn't require any class or function. You can use one or more declarations or expressions that are evaluated and produce output.

**For example, the script below:**

```
var firstdigit : Integer = 10;
var seconddigit : Integer = 20;
var sum : Integer = firstdigit + seconddigit;
java.lang.System.out.println("Sum of {firstdigit} and {seconddigit} is {sum}.");
```

**Produces output as:**

Sum of 10 and 20 is 30.

## ii) Classes and objects:

### 1. Importing Classes:

Importing packages and classes is like the way we do while programming with java.

```
import MyPackage1.*;
import MyPackage2.MyClass;
```

In JavaFX Script programming language, standard java programming language packages and classes can also be imported along with its built-in packages and classes.

**For example:**

```
import javafx.animation.*;
import javafx.scene.image.Image;
import java.lang.System;
```

### 2. Defining Classes:

The class in JavaFX can be defined with 'class' keyword. The 'extends' keyword can be used for inheritance purpose. JavaFX supports multiple inheritance so you can list the name of base classes separated with comma after the 'extends' keyword.

**For example:**

```
public class AnimatedImage extends ImageView {
.....
.....
    private attribute imagesCount: Integer = 0;
    private attribute images: Image[];
    private attribute currentIndex: Integer ;
```

## Working with JavaFX Classes and Objects

```
.....  
.....  
public function play():Void {  
    if (images == null) {  
        loadImages();  
    }  
    movieTimeline.start();  
}  
public function stop():Void {  
    movieTimeline.stop();  
}  
.....  
.....  
}
```

### 3. Creating Objects:

There are two ways to instantiate the class: Using object literal and the traditional way i.e. using 'new' keyword. Object literal should be preferred for creating object. For this, write the name of the class followed by pair of curly braces and then inside of the braces give values to the attributes of the class.

#### For example:

```
class MyClass {  
    attribute firstvalue: Integer;  
    attribute secondvalue: Integer;  
    function add(value1: Integer, value2: Integer): Integer {  
        return value1 + value2;  
    }  
}  
//Create object of MyClass  
var obj = MyClass{  
    firstvalue : 10  
    secondvalue : 20  
}  
var a = obj.firstvalue + 5;  
var b = obj.secondvalue + 5;  
java.lang.System.out.println("Sum is: "+ obj.add(a,b));
```

The output of the above program will be as below:

```
Sum is: 40
```

### iii) Attributes and Functions:

#### 1. Attributes:

Attributes represent the states of the object. Attributes are declared using 'attribute' keyword as given below:

# Working with JavaFX Classes and Objects

```
attribute firstvalue: Integer;  
attribute secondvalue: Integer;  
attribute thirdvalue: Number = 15;
```

Attribute's value can be assigned at the place of declaration. If they are not initialized explicitly the default value is assigned.

## 2. Functions:

Functions represent behavior of the object. Its syntax is:

```
function functionName (parameterName : parameterType, .....): returnType function_body
```

### For example:

```
function add(value1: Integer, value2: Integer): Integer {  
    return value1 + value2;  
}
```

## iv) Variables and Basic Data Types

### 1. Variables

To create a variable in the program 'var' keyword is used. Its syntax can be as below:

```
var variableName : type = initializer;
```

### For example:

```
var title = "JavaJazzUp";  
var variable1 = 1;  
var variable2 : Number = 2;
```

### 2. Basic Data Types

JavaFX has five basic datatypes:

1. String (as **java.lang.String** in Java programming language)
2. Boolean (as **java.lang. Boolean** in Java programming language)
3. Number (as **java.lang. Number** in Java programming language)
4. Integer (as **byte,short,int,long,BigInteger** in Java programming language)
5. Duration

The Duration type is one of the new type in JavaFX. This class represents a unit of time (millisecond, second, minute, or hour). We can instantiate this class using time literal also.

```
1ms; // 1 milliseconds  
2s; // 2 seconds  
3m; // 3 minutes  
4h; // 4 hour
```

# Working with JavaFX Classes and Objects

## v) Using 'if' expression:

Using 'if' expression is similar to the java programming language.

### The syntax is like as below:

```
if (condition) {
    .....
    //Statement1
    //Statement2
    .....
} else if (condition) {
    .....
    //Statement1
    //Statement2
    .....
} else {
    .....
    //Statement1
    //Statement1
    .....
}
```

## vi) Using while expression:

Using while expression is similar to the java programming language but curly braces are required with this statement.

### The syntax for using while:

```
while (condition){
.....
    //Statement1
    //Statement2
.....
}
```

## vii) Operators:

Most of the JavaFX operators are as in Java Programming Language. In the table below, operators are listed by precedence and also compared to the Java Programming Language.

### Operator Precedence

Priority	JavaFX Script Programming Language Operator	Operation	Java Programming Language Operator	Order of Evaluation
1	function() ( )	JavaFX function An expression in	function()	Class n/a



## Working with JavaFX Classes and Objects

	new	brackets		Class
	{Object literal}	Instantiate a new object		Class
2	++ (suffix)	Instantiate and initialize a new object		
	— (suffix)	Post-increment assign	++	Right to Left
3	++ (prefix)	Post-decrement assign	—	
	— (prefix)	Pre-increment assign	++	Right to Left
	not	Pre-decrement assign	—	
	sizeof	Logical negation	<>	Boolean
	reverse	Size of a sequence		Sequence
	indexof	Reverse of a sequence		
		Index of a sequence element		
4	—>	tween		
	*	Multiplication	*	Left to Right
	/	Division	/	
5	mod	Remainder	%	
	+	Addition	+	Left to Right
	-	Subtraction	-	
6	==	Equality	==	Left to Right
	!=	Inequality	!=	
	<	Less than	<	
	<=	Less than or equal	<=	
	>	Greater than	>	
	=>	Greater than or equal to	=>	
7	instanceof	Type checking	Instanceof	Class
	as	Cast		Class
8	or	Logical OR		Right to Left
9	and	Logical AND	&&	Right to Left
10	+=	Add and assign	+=	
	-=	Subtract and assign	-+	
	*=	Multiply and assign	*=	
	/=	Divide and assign	/=	
	%=	Remainder and assign	%=	
11	=	Assignment	=	(all)

### viii) Keywords and Reserved Words:

In the table below, all the Implemented and reserved keywords have been listed.

# Working with JavaFX Classes and Objects

## Implemented Keywords

abstract	after	and
as	at	attribute
before	bind	bound
break	catch	class
continue	dur	delete
else	exclusive	extends
false	finally	for
function	if	in
indexof	init	insert
into	inverse	instanceof
import	lazy	new
not	null	or
on	override	package
private	public	return
replace	super	sizeof
static	step	then
this	throw	true
try	tween	var
with	where	while

## Reserved Keywords

assert	by	do
first	from	last
lazy	let	protected
readonly	typeof	

### ix) Sequences:

Sequences in JavaFX are like arrays in Java Programming Language.

#### Sequences are declared as:

```
var myArray:Integer[]; // Sequence of Integer
var myStringArray:String[]; // Sequence of String
```

#### For example:

```
var myArray:Integer[];
myArray =[10,20];
java.lang.System.out.println("Array elements are:");
java.lang.System.out.println(myArray [0]);
java.lang.System.out.println(myArray [1]);
```

Output of the above code will be as below:

```
Array elements are:
10
20
```

## Working with JavaFX Classes and Objects

**The sequence can also be created as below:**

```
var numbers = [10,20,30,40];  
var names = ["Java", "FX"];
```

Sequences are not objects but the ordered list of objects. If their length and elements are equal then they are considered to be equal.

**For example, the code:**

```
var myArray:Integer[];  
myArray=[10,20];  
java.lang.System.out.println("Equal? "+ {myArray == [10,20]});
```

**Gives output as below:**

```
Equal? true
```

It has one interesting feature: It provides a short notation '..' for sequences in arithmetic series.

**For example, the code below:**

```
var values = [1..50]; // No need to type all numbers from 1 to 50.  
  
java.lang.System.out.println("values[0]: "+ values[0]);  
java.lang.System.out.println("values[1]: "+ values[1]);  
java.lang.System.out.println("values[2]: "+ values[2]);  
java.lang.System.out.println("values[49]: "+ values[49]);
```

**Gives output as below:**

```
values[0]: 1  
values[1]: 2  
values[2]: 3  
values[49]: 50
```

**Inserting an element into a sequence:**

Syntax for inserting a new element in the sequence is like below:

```
insert n into mySequence;  
insert n before mySequence [index];  
insert n after mySequence [index];
```

**For example, the code below:**

```
var mySequence = [1..5];  
insert 7 into mySequence;  
insert 6 before mySequence [5];
```

## Working with JavaFX Classes and Objects

```
insert 8 after mySequence [6];
for (i in mySequence){
    java.lang.System.out.println("value: "+i);
}
```

**Produces output as below:**

```
value: 1
value: 2
value: 3
value: 4
value: 5
value: 6
value: 7
value: 8
```

**Deleting an element from a sequence:**

```
delete mySequence;
delete n from mySequence;
delete mySequence [index];
delete mySequence [n..m];
```

For example, the code below:

```
var mySequence = [1..10];
delete 1 from mySequence;
for (i in mySequence){
    java.lang.System.out.print(i);
}
java.lang.System.out.println("");
delete mySequence [1];
for (i in mySequence){
    java.lang.System.out.print(i);
}
java.lang.System.out.println("");
delete mySequence [2..4];
for (i in mySequence){
    java.lang.System.out.print(i);
}
java.lang.System.out.println("");
delete mySequence;
for (i in mySequence){
    java.lang.System.out.print(i);
}
```

**Produce output as below:**

```
2345678910
245678910
248910
```

## Working with JavaFX Classes and Objects

### Iterating sequence using 'for' loop:

Sequence can be iterated using 'for' operator.

### For example, the code below:

```
var values = [1..5];
for (i in values){
    java.lang.System.out.println("value: "+i);
}
```

### Gives output as below:

```
value: 1
value: 2
value: 3
value: 4
value: 5
```

### xi) Data Binding:

Data binding is one of the important feature of JavaFX. The general syntax of using 'bind' is:

```
var myVariable = bind myExpression;
```

In the above syntax, variable 'myVariable' is bound to the value of the expression 'myExpression'. As the value of the expression 'myExpression' changes, the changes also reflect to the value of the variable 'myVariable'.

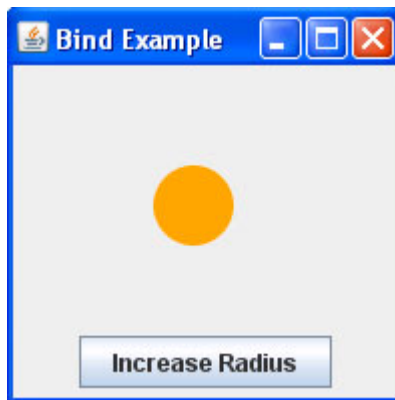
For example, in the code below the radius of the circle is bound to the expression  $(X+Y)/2$ . When the expression value is changed due to changes in value of the variables X Or Y, the radius of the circle also changes.

```
import javafx.scene.geometry.Circle;
import javafx.scene.paint.Color;
import javafx.ext.swing.*;
var x = 20;
var y = 20;
SwingFrame {
    title: "Bind Example"
    width: 200
    height: 200
    closeAction: function(): Void { java.lang.System.exit( 0 ); }
    visible: true
    content : BorderPanel {
        top: Canvas {
            content: [Circle {
                centerX: 90, centerY: 70;
                radius: bind (x+y)/2;
                /* Now the radius of the circle is bound to the expression (x+y)/2. As the value of the
                expression changes, the radius of the circle also changes. */
                fill: Color.ORANGE
            } ]
        }
    }
}
```



## Working with JavaFX Classes and Objects

```
}  
bottom : FlowPanel {  
    content:[Button {  
        text: "Increase Radius"  
        action: function(): Void {  
            x++;  
            y++;  
        }  
    }]  
}
```



# Developing “Hello World Example” in JavaFX

## JavaFX script “Hello World” Example

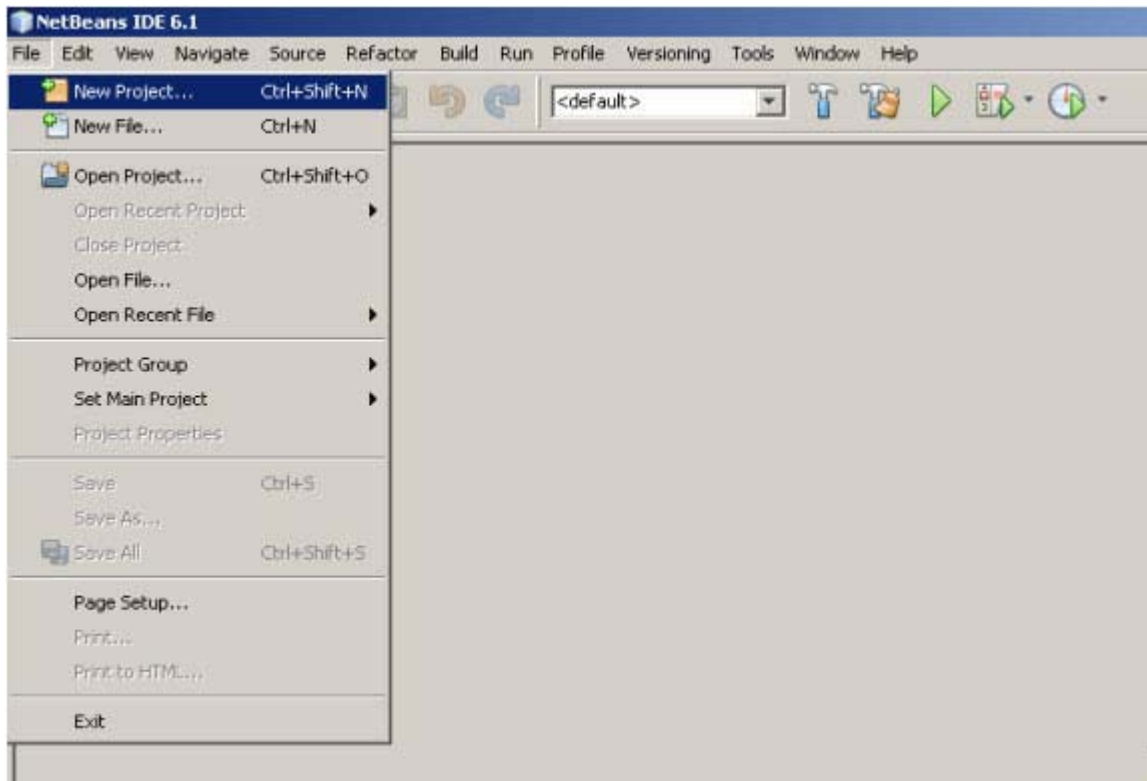
Following are the step-by-step procedure to create first “Hello World” example

### Step: -1 Open Net Beans IDE



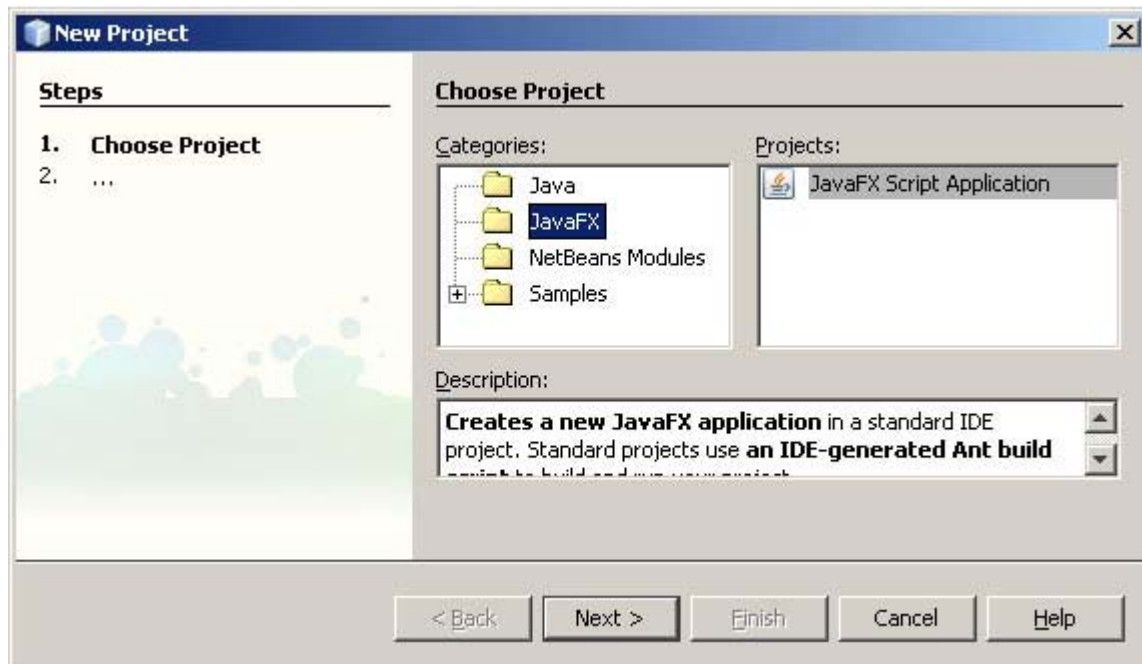
### Step: -2 Open New Project

# Developing “Hello World Example” in JavaFX




# Developing “Hello World Example” in JavaFX

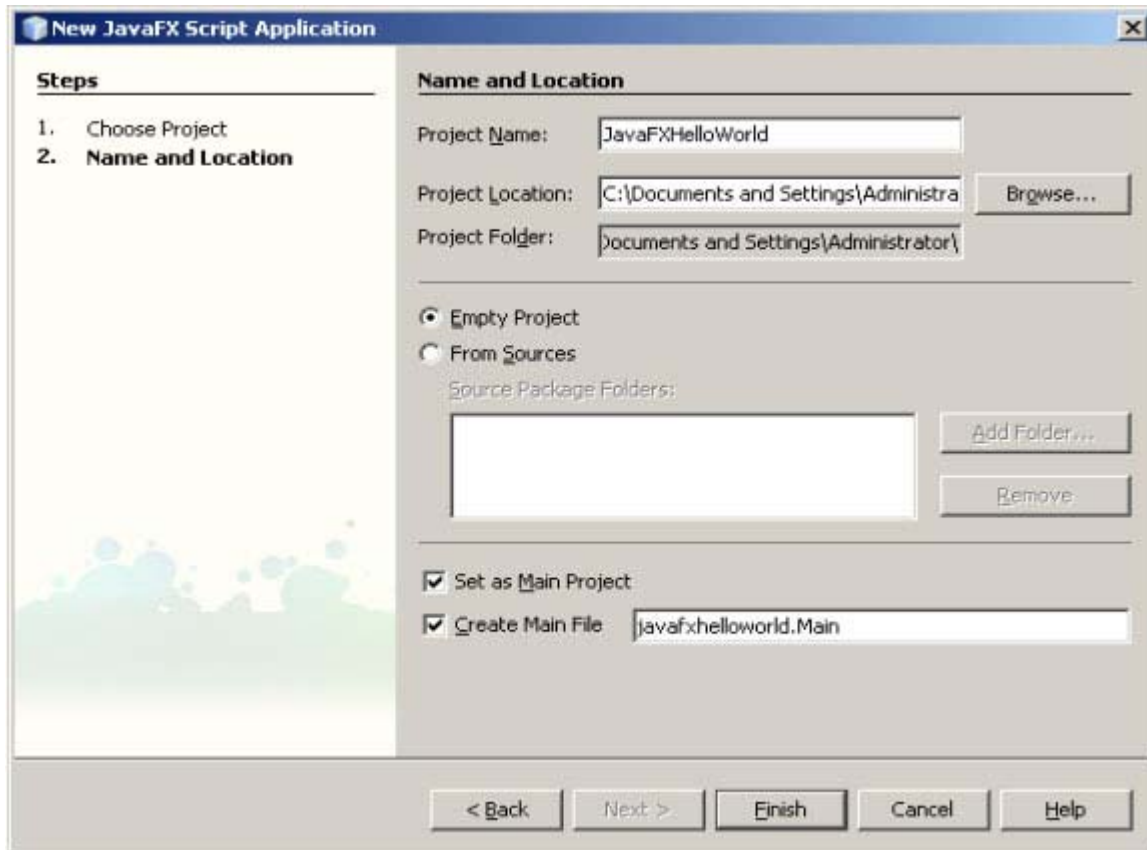
## Step: - 3 Open JavaFX Script Application



Click on 

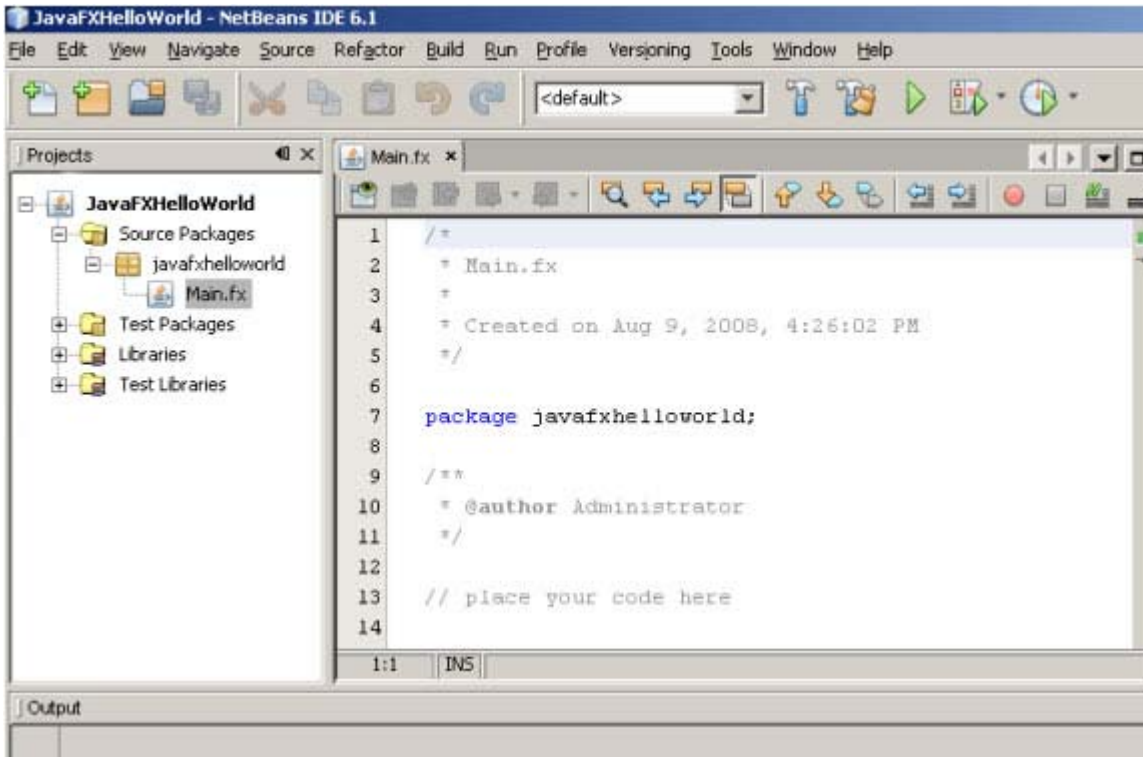
Step: - 4 Create Project with name as you want and Click on 

## Developing “Hello World Example” in JavaFX



**Step:-5 Type your program here**

# Developing “Hello World Example” in JavaFX



**Step: - 6 Write following code in Main.fx file.**

```
/*
 * Main.fx
 *
 * Created on Aug 9, 2008, 4:26:02 PM
 */
```

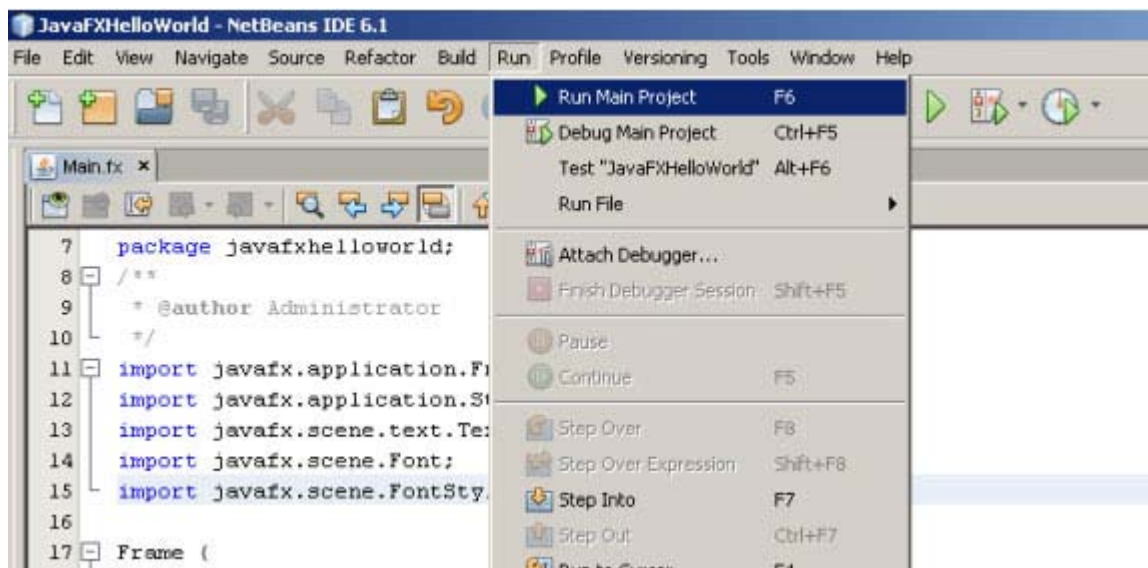
```
package javafxhelloworld;
/**
 * @author Administrator
 */
import javafx.application.Frame;
import javafx.application.Stage;
import javafx.scene.text.Text;
import javafx.scene.Font;
import javafx.scene.FontStyle;
```

```
Frame {
    title:"Java FX Example"
    width:300
    height:300
    visible:true
    stage:Stage {
```

## Developing “Hello World Example” in JavaFX

```
content: [ Text {  
    font: Font {  
        size: 24  
        style: FontStyle.PLAIN  
    }  
    x: 10, y: 30  
    content: "HelloWorld"  
} ]  
}
```

### Step: - 7 Run Project



### Step: - 8 Output of your program would be like this



## Developing “Hello World Example” in JavaFX



**[Download Source Code](#)**

# Advanced Examples to JavaFX

After developing Hello application, we are now going to create some more examples having different features. We will develop examples:

1. Text Animation Example
2. Circle Moving Example
3. Clock in JavaFX
4. Graphics Example
5. Creating Progress Bar
6. Media Player Example
7. Login Application
8. Applet Example
9. Creating Slider

## (I) Animation Examples

1. Text Animation Example
2. Circle Moving Example

### 1. Text Animation Example

To create Animation in JavaFX Script we have to use **"javafx.animation"** package and its classes.

```
package javafxanimation;

/**
 * @author Amit
 */

import javafx.ext.swing.*;
import javafx.scene.geometry.*;
import javafx.scene.paint.*;
import javafx.animation.*;
import java.lang.System;
import javafx.scene.text.Text;
import javafx.scene.Font;
import javafx.scene.FontStyle;

var x = 0;
var t = Timeline {
    repeatCount: 100
    autoReverse: true

    keyFrames: [KeyFrame{time : 0s
                        values: x => 0},

                KeyFrame{time : 2s
                        values: x => 100 tween Interpolator.EASEBOTH}
    ]
}
SwingFrame {
    title: "Animation"
    width: 250
}
```

## Advanced Examples to JavaFX

```
height: 200
closeAction: function(): Void { java.lang.System.exit( 0 ); }
visible: true
content : BorderPanel {
    top: Canvas {
        content: [ Text {
            font: Font {
                size: 24
                style: FontStyle.PLAIN
            }
            x: bind x, y:20;
            content: "Hello World"
            fill:Color.BLUE
        }]
    }

    bottom : FlowPanel { content:[Button {
        text: "Start"
        action: function(): Void {t.start();}
    },
    Button {
        text: "Stop"
        action: function():Void {t.stop(); }
    }]
    }
}
}
```

### Output:



By clicking on Start button Animation starts and **"Hello World"** moves in Horizontal direction.

# Advanced Examples to JavaFX



Stop will stop animation.

## Download Source Code

### 2. Circle Animation

To create animation with shape circle we have used "javafx.scene.geometry" package and "javafx.animation" package.

```
package javafxcircle;

import javafx.scene.geometry.Circle;
import javafx.scene.paint.Color;
import javafx.animation.Timeline;
import javafx.animation.KeyFrame;
import javafx.animation.Interpolator;
import javafx.ext.swing.*;

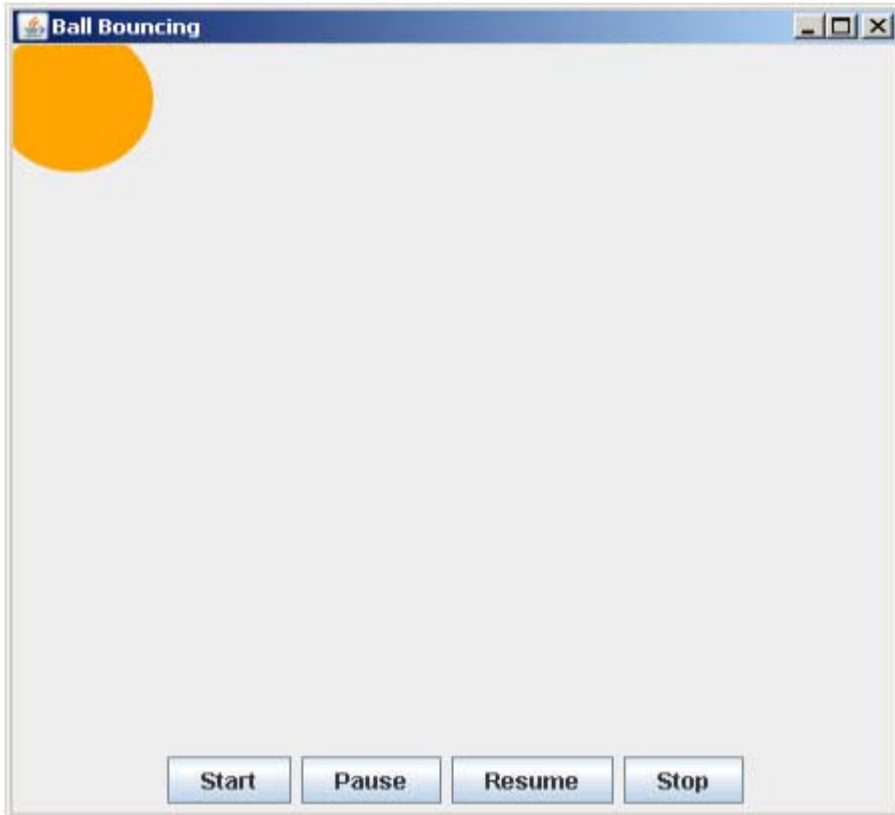
/**
 * @author Amit
 */
var x = 30;
var y = 30;
var t = Timeline {
    repeatCount: Timeline.INDEFINITE
    autoReverse: true
    keyFrames : [
        KeyFrame {time : 0s
            values: y => 30},
        KeyFrame {time: 2s
            values: y => 400 tween Interpolator.LINEAR},
        KeyFrame {time : 1s
            values: x => 30},
        KeyFrame {time: 3s
            values: x => 400 tween Interpolator.EASEIN},
        KeyFrame {time : 2s
            values: y => 400},
        KeyFrame {time: 4s
```

## Advanced Examples to JavaFX

```
        values: y => 0 tween Interpolator.EASEBOTH}
    ]
}
SwingFrame {
title: "Ball Bouncing"
width: 450
height: 450
closeAction: function(): Void { java.lang.System.exit( 0 ); }
visible: true
content : BorderPanel {
top: Canvas {
content: [Circle {
centerX: bind x, centerY: bind y;
radius: 40
fill: Color.ORANGE
} ]
}
bottom : FlowPanel { content:[Button {
text: "Start"
action: function(): Void {t.start();}
},Button {
text: "Pause"
action: function() {t.pause(); }},Button {
text: "Resume"
action: function() {t.resume();}
},Button {
text: "Stop"
action: function() {t.stop();}
}
]
}
}
}
```

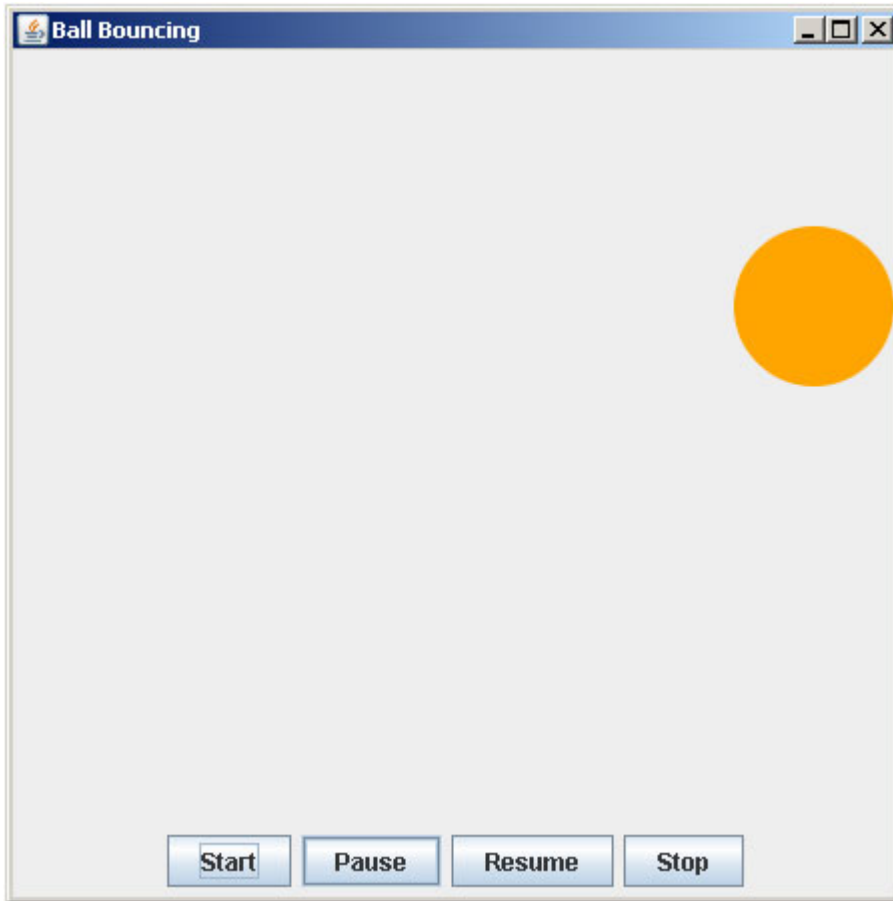
## Advanced Examples to JavaFX

Output:



By Clicking onstart button will start animation

## Advanced Examples to JavaFX



Pause Button will pause animation.

[Download Source Code](#)

### (II) Clock in JavaFx

To create Clock in JavaFX we have created a class Clock in Main.fx file and it creates a custom node by extending "CustomNode" class in "javafx.scene" package.

```
package javafxclock;

/**
 * @author Amit
 */

import javafx.scene.Node;
import javafx.scene.CustomNode;
import javafx.scene.Group;
import javafx.scene.geometry.Circle;
```



## Advanced Examples to JavaFX

```
import javafx.scene.geometry.Line;
import javafx.application.Frame;
import javafx.application.Stage;
import javafx.scene.paint.Color;
import javafx.scene.transform.Translate;
import javafx.scene.transform.Rotate;
import javafx.animation.Timeline;
import javafx.animation.KeyFrame;
import java.util.Calendar;
import java.lang.System;
import javafx.scene.image.ImageView;
import javafx.scene.image.Image;

var clock : Clock = Clock {};
clock.timer.start();

Frame {
    stage : Stage {
        content : clock
    }

    visible : true
    title : "JavaFXClock"
    width : 200
    height : 230
    closeAction : function() {
        java.lang.System.exit( 0 );
    }
}

public class Clock extends CustomNode {

    attribute seconds : Number;
    attribute minutes : Number;
    attribute hours : Number;

    public attribute timer : Timeline = Timeline {
        repeatCount : Timeline.INDEFINITE
        keyFrames :
            KeyFrame {
                time : 1s
                action : function() {
                    var calendar : Calendar = Calendar.getInstance();
                    seconds = calendar.get( Calendar.SECOND );
                    minutes = calendar.get( Calendar.MINUTE );
                    hours = calendar.get( Calendar.HOUR_OF_DAY );
                }
            }
    };

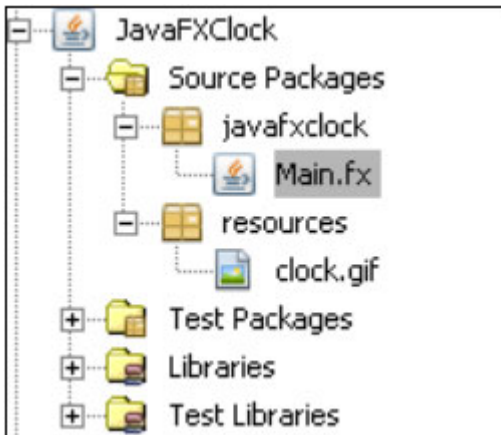
    public function create(): Node {
        return Group {
```

## Advanced Examples to JavaFX

```
content : [
  ImageView {
    transform: [ ]
    image: Image {url: "{__DIR__}/../resources/clock.gif"}
  },
  Line {
    transform : [ Rotate { angle : bind seconds * 6, x : 100, y : 100 } ]
    startX : 100
    startY : 30
    endX : 100
    endY : 100
    stroke : Color.WHITE
  },
  Line {
    transform : [ Rotate { angle : bind minutes * 6, x : 100, y : 100 } ]
    startX : 100
    startY : 40
    endX : 100
    endY : 100
    stroke : Color.LIGHTGRAY
    strokeWidth : 2
  },
  Line {
    transform : [ Rotate { angle : bind hours * 30, x : 100, y : 100 } ]
    startX : 100
    startY : 50
    endX : 100
    endY : 100
    stroke : Color.DARKGRAY
    strokeWidth : 4
  }
]
};
}
```

**Project tree structure would be like this:**

## Advanced Examples to JavaFX



**Output:**



**Download Source Code**

### (III) JavaFX Graphics

We can handle Graphics with "**javafx.scene.geometory**" package.

```
/*  
 * Example.fx  
 *  
 * Created on Aug 8, 2008, 4:11:25 PM  
 */
```

## Advanced Examples to JavaFX

```
package javafxgraphics;

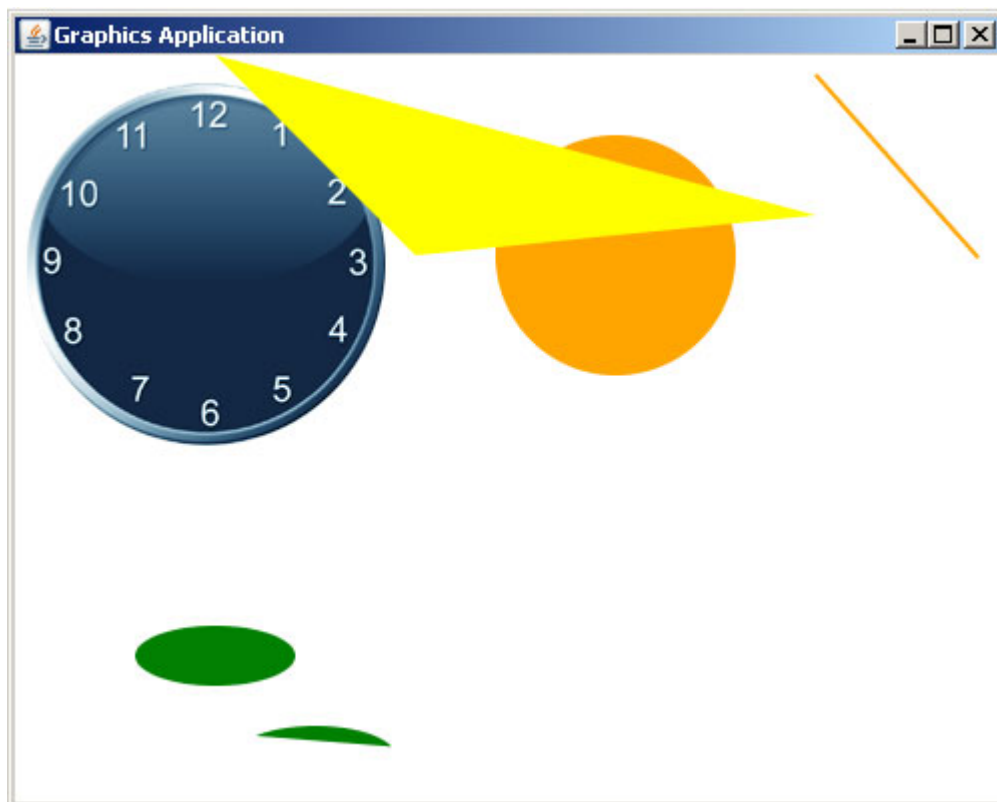
import javafx.application.Stage;
import javafx.application.Frame;
import javafx.ext.swing.GridPanel;
import javafx.scene.geometry.Circle;
import javafx.scene.paint.Color;
import javafx.scene.geometry.Arc;
import javafx.scene.geometry.ArcType;
import javafx.scene.geometry.Ellipse;
import javafx.scene.image.ImageView;
import javafx.scene.image.Image;
import javafx.scene.geometry.Line;
import javafx.scene.geometry.Polygon;

/**
 * @author Amit
 */
var x=300;
var y=100;
var component ; //Component Variable
var circle=Circle {
    centerX: bind x, centerY: bind y;
    radius: 60
    fill: Color.ORANGE
}
var arc=Arc {
    centerX: 150, centerY: 350
    radiusX: 40, radiusY: 15
    startAngle: 18, length: 120
    type: ArcType.OPEN
    fill: Color.GREEN
}
var ellipse=Ellipse {
    centerX: 100, centerY: 300
    radiusX: 40, radiusY: 15
    fill: Color.GREEN
}
var image=ImageView {
    image: Image {
        url: "{__DIR__}/../resources/clock.gif"
    }
}
var line=Line {
    startX: 400, startY: 10
    endX: 480, endY: 100
    strokeWidth: 2
    stroke: Color.ORANGE
}
var poly=Polygon {
    points : [ 200,100, 100,0, 400,80 ]
    fill: Color.YELLOW
```

## Advanced Examples to JavaFX

```
}  
Frame {  
  title: "Graphics Application"  
  width: 500  
  height: 400  
  closeAction: function() {  
    java.lang.System.exit( 0 );  
  }  
  visible: true  
  stage : Stage {  
    content: [ arc,  
              circle,  
              image,  
              line,  
              ellipse,  
              poly  
            ]  
  }  
}
```

**Output:**



# Advanced Examples to JavaFX

## Download Source Code

### (IV) Creating Progress Bar in JavaFX script

We can create Progress Bar in JavaFX script with the help of Rectangle and **"javafx.scene.geometry"** and **"javafx.animation"** package.

```
package javafxprogressbar;
/**
 * @author Amit
 */

import javafx.ext.swing.*;
import javafx.animation.*;
import javafx.scene.Font;
import javafx.scene.FontStyle;
import javafx.scene.text.Text;
import javafx.scene.paint.Color;
import javafx.ext.swing.Button;
import javafx.animation.Timeline;
import javafx.animation.KeyFrame;
import javafx.ext.swing.SwingFrame;
import javafx.ext.swing.BorderPanel;
import javafx.scene.geometry.Rectangle;

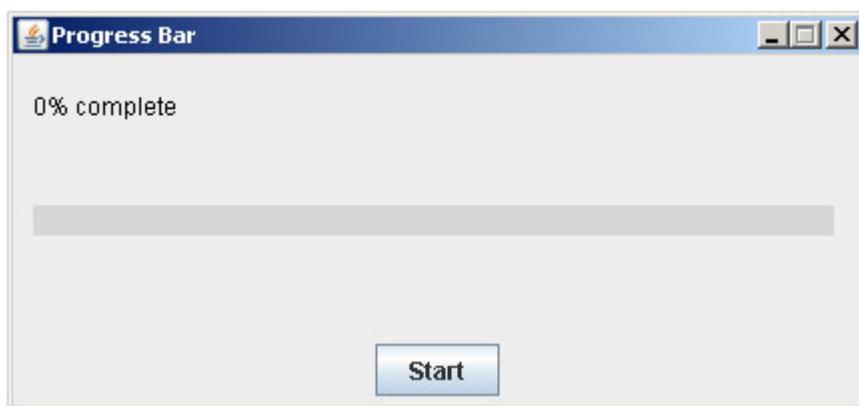
var wdth=0;
var t=Timeline {
    repeatCount: 1
    keyFrames: [KeyFrame{time : 0s
                        values: wdth => 0},

                KeyFrame{time : 4s
                        values: wdth => 400 tween Interpolator.EASEBOTH }
    ]
}
SwingFrame {
    title: "Progress Bar"
    width: 430
    height: 200
    resizable:false
    closeAction: function() {
        java.lang.System.exit(0);
    }
    visible: true
    content: BorderPanel {
        top:Canvas{
            content:[Text {
                font: Font {
                    size: 12
                    style: FontStyle.PLAIN
                }
            }
        ]
        x: 10, y: 30
    }
}
```

## Advanced Examples to JavaFX

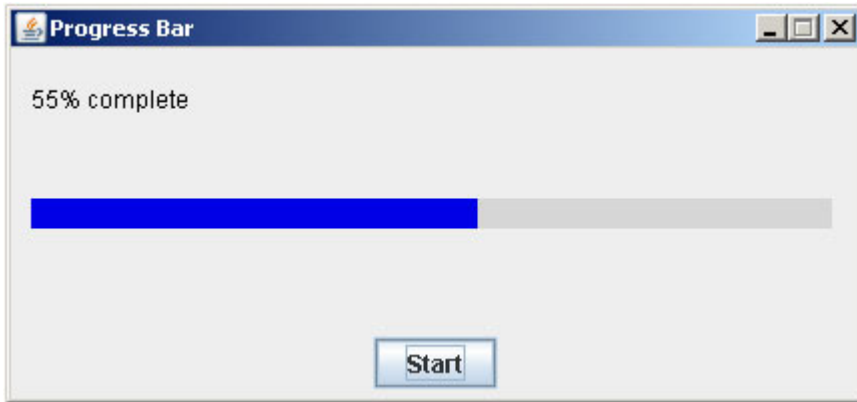
```
        content: bind "{width/4}% complete"
    }]
}
center: Canvas{
    content: [
        Rectangle {
            x: 10, y: 40
            width:bind width, height: 15
            fill: Color.BLUE
            opacity:1
        },Rectangle {
            x: 10, y: 40
            width: 400, height: 15
            fill: Color.BLACK
            strokeWidth:2
            opacity:0.1
        }
    ]
}
bottom: FlowPanel{
    content: [Button {
        text: "Start"
        action: function() {
            t.start();
        }
    }]
}
}
```

**Output:**

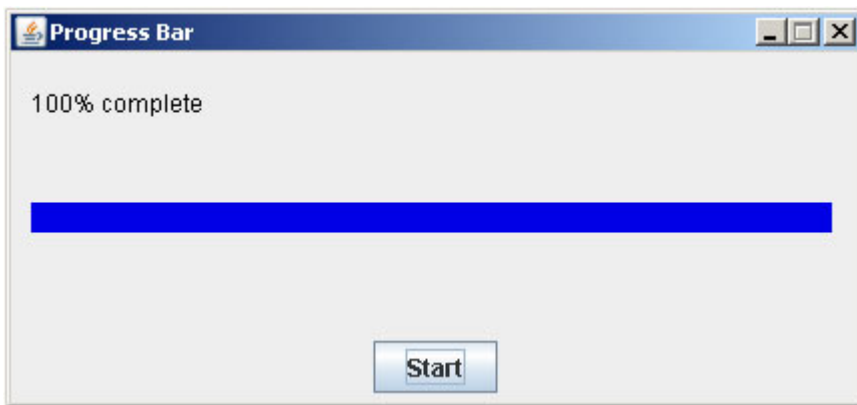


**By Clicking on Start Button will start Progress bar progress.**

## Advanced Examples to JavaFX



**On 100 % complete progress will stop.**



Download Source Code

### **(V) JavaFX Media Player**

We can create Media Player with the use of `javafx.scene.media` package.

```
package javafxmediaplayer;

import javafx.scene.media.*;
import javafx.ext.swing.SwingFrame;
import javafx.ext.swing.BorderPanel;
import javafx.ext.swing.FlowPanel;
import javafx.ext.swing.Canvas;
import import javafx.ext.swing.Button;
import javafx.input.MouseEvent;
import javafx.scene.image.ImageView;
import javafx.scene.image.Image;

/**
```



## Advanced Examples to JavaFX

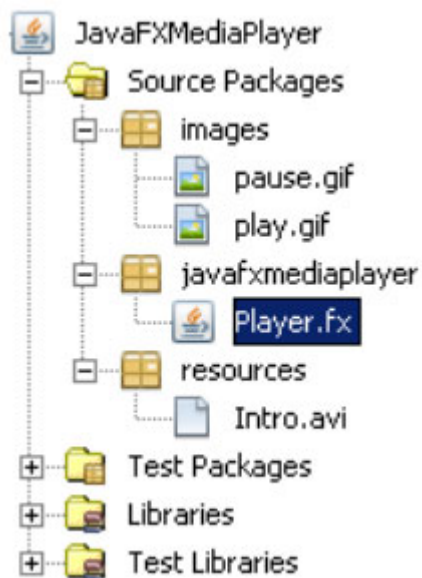
```
* @author Amit
*/
var mediaplyr = MediaPlayer {
    autoPlay: false
    repeatCount: 3
    media: bind Media {
        source: "{__DIR__}/../resources/Intro.avi"
    }
}
var image1= Image {
    url: "{__DIR__}/../images/play.gif"
}
var image2= Image {
    url: "{__DIR__}/../images/pause.gif"
}
var opaque:Number=1;
var transparent:Number=0.5;
var opacityPlay:Number = transparent;
var opacityPause:Number = transparent;

SwingFrame {
    title: "Media Application"
    width: 650
    height: 550
    resizable:false
    closeAction: function() {
        java.lang.System.exit( 0 );
    }
    visible: true
    content : Canvas{
        content :[MediaView {
            mediaPlayer: mediaplyr
        },ImageView {
            image:image1
            x:20
            y:420
            opacity: bind opacityPlay;
            onMouseMoved: function( e: MouseEvent ):Void {
                opacityPlay = opaque;
            }
            onMouseExited: function( e: MouseEvent ):Void {
                opacityPlay = transparent;
            }
            onMouseClicked: function( e: MouseEvent ):Void {
                mediaplyr.play();
            }
        },ImageView {
            image: image2
            x:520
            y:420
            opacity: bind opacityPause;
            onMouseMoved: function( e: MouseEvent ):Void {
```

## Advanced Examples to JavaFX

```
        opacityPause = opaque;    }
    onMouseExited: function( e: MouseEvent ):Void {
        opacityPause = transparent;    }
    onMouseClicked: function( e: MouseEvent ):Void {
        mediaPlayer.pause();
    }
}
]
```

**Project tree structure would be like this:**



## Advanced Examples to JavaFX

**Output:**



When we click Mouse over play and pause button then it will **Play** and **Pause** Media player.

**Download Source Code**

**(Vi) Login Application in JavaFX script**

```
package javafxlogin;

import javafx.ext.swing.*;
import javafx.ext.swing.Label;
import javafx.ext.swing.SwingFrame;
import javafx.ext.swing.Canvas;
import javafx.ext.swing.MenuItem;
import javafx.ext.swing.FlowPanel;
import javafx.ext.swing.GridPanel;
import javafx.ext.swing.TextField;
import javafx.scene.text.Text;
import javafx.scene.Font;
import javafx.scene.FontStyle;
import javafx.scene.paint.Color;

/**
 * @author Amit
 */
```

## Advanced Examples to JavaFX

```
var firstname:String="";
var lastname:String="";
var laberror= Label {
    text: "Fill up all entries properly"
    foreground:Color.RED
    visible:false
}
var loginLabel=Label {
    text:"Login Page"
}
var loginFrame = SwingFrame {
    title: "Welcome User"
    width: 250
    height: 200
    closeAction: function() {
        java.lang.System.exit( 0 );
    }
    visible: false
    resizable :false
    content:BorderPanel {
        center:Label {
            text: "\\t \\t Welcome \\tAmit \\t :-)"
        }
    }
}

var frame = SwingFrame {
    resizable:false
    title: "Login Application"
    width: 250
    height: 200
    closeAction: function() {
        java.lang.System.exit( 0 );
    }
    visible: true
    content: BorderPanel{
        top: loginLabel
        background:Color.ORANGE
        var text1 = TextField {
            columns: 10
            text: ""
            editable: true
            selectOnFocus:true
        }
        var text2 = TextField {
            columns: 10
            text: ""
            editable: true
            selectOnFocus:true
        }
        var userLabel=Label {
            text: "FirstName"
```

## Advanced Examples to JavaFX

```
}
var passLabel=Label {
    text: "LastName"
}

var submit = Button {
    text: "Submit"
    action: function() {
        firstname = text1.text;
        lastname = text2.text;
        if(firstname.equals("") or lastname.equals(""))
        {
            laberror.visible=true;
        }
        else
        {
            laberror.text="Incorrect firstname or lastname !";
            laberror.visible=true;
        }

        if(firstname.equals("Amit") and lastname.equals("Kumar")){
            loginFrame.visible=true;
        }
    }
}

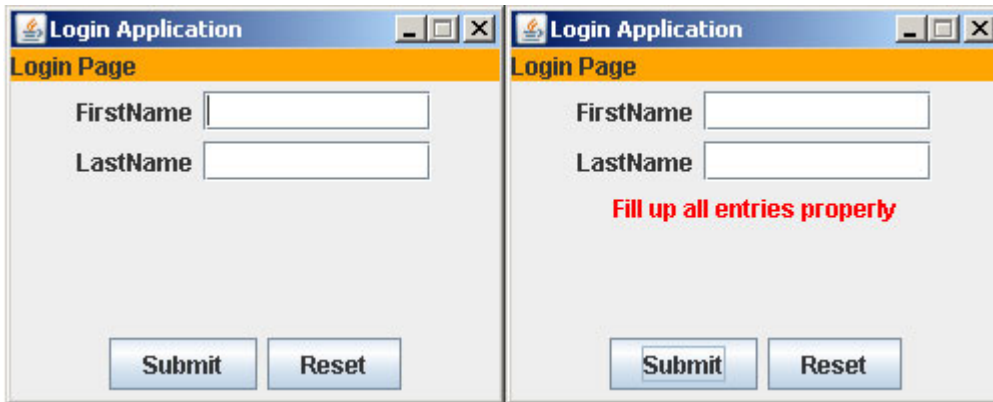
var panel= FlowPanel{
    content:[ submit,Button {
        text: "Reset"
        action: function() {
            text1.text="";
            text2.text="";
            laberror.visible=false;
        }
    }
    ]
}

center: FlowPanel{
    content:[ userLabel,
        text1, passLabel , text2, laberror
    ]
}
bottom: panel
}
```

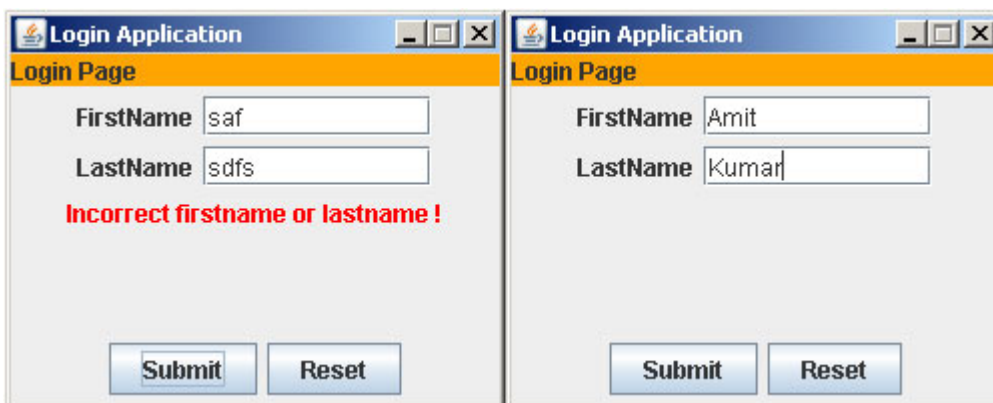
### Output:

If we "Click" on submit button without filling proper entries that it will show message "Fill Up all entries properly"

## Advanced Examples to JavaFX



If either of text field **FirstName** or **LastName** is incorrect then it will flash message “**Incorrect firstname or lastname !**”



When we enter correct values in **FirstName** and **LastName** fields then it will show another “**Welcome**” frame.



# Advanced Examples to JavaFX

## Download Source Code

### (VII) JavaFX Applet Example

We can also create Applet Application with the use of JavaFX Script. To create applet we have to create JavaFX Applet File.

#### **applet.fx**

```
/*
 * applet.fx
 *
 * Created on Aug 7, 2008, 1:08:44 PM
 */
package javafxapplet;

import javafx.application.*;
import javafx.scene.*;
import javafx.scene.text.*;

/**
 * @author Amit
 */
Application{
    stage: Stage{
        content: Text {
            x: 10, y: 30
            content: "Application content"
            font: Font { size: 24 style: FontStyle.PLAIN }
        }
    }
}
```

#### **appletHtml.html**

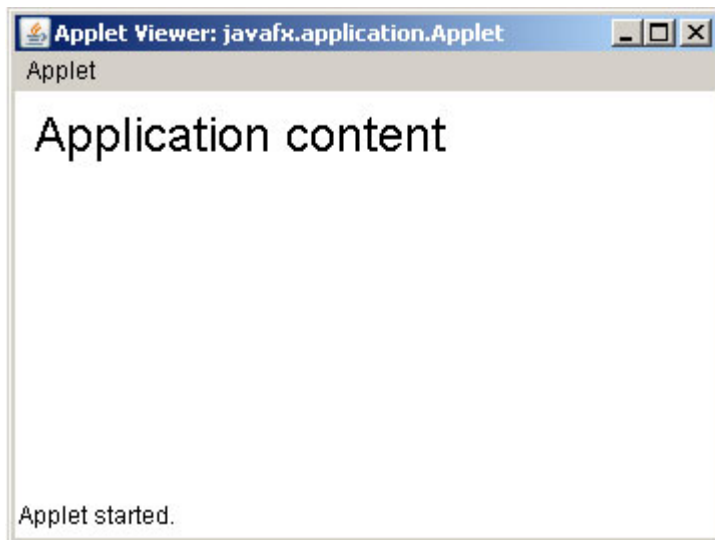
We can also add this applet file in HTML page with the following code:

```
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<applet height="300" width="400" archive="lib/javafxrt.jar,JavaFXApplet.jar"
code="javafx.application.Applet">
<param name="ApplicationClass" value="javafxapplet.applet"/>
</applet>
</body>
</html>
```

## Advanced Examples to JavaFX

### Output:

To run applet, use "Run Applet" command. Right click on applet file in project tree and click on "Run Applet".



[Download Source Code](#)

### (VIII) JavaFX Slider Example

We can create Slider with JavaFX .

```
package javafxslider;

import javafx.ext.swing.SwingFrame;
import javafx.ext.swing.BorderPanel;
import javafx.ext.swing.Slider;
import javafx.ext.swing.Label;
import javafx.scene.paint.Color;

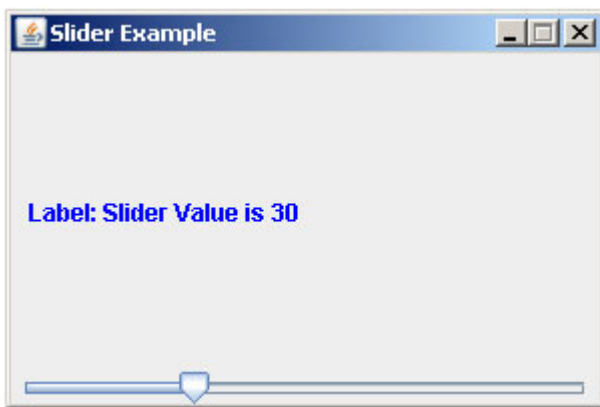
/**
 * @author Amit
 */
var hght:Integer=30;

SwingFrame {
    title: "Slider Example"
    width: 300
    height: 200
    resizable:false
    closeAction: function() {
        java.lang.System.exit( 0 );
    }
    visible: true
```



## Advanced Examples to JavaFX

```
content: BorderPanel {  
    center: Label {  
        text: bind "\t \t \t Label: Slider Value is {hght}"  
        foreground:Color.BLUE  
    }  
    bottom: Slider {  
        minimum: 0  
        maximum: 100  
        value : bind hght with inverse  
    }  
}  
}
```



[Download Source Code](#)

### (IX) Images Example in JavaFX Script

We can show and handle images in JavaFX also by using "javafx.scene.image" package.

**package javafxpictures;**

```
import javafx.ext.swing.*;  
import javafx.scene.paint.*;  
import javafx.scene.image.*;  
import javafx.ext.swing.SwingFrame;  
import javafx.ext.swing.Canvas;  
import javafx.scene.image.ImageView;  
import javafx.ext.swing.Layout.Alignment;  
import javafx.scene.image.Image;  
import java.lang.System;  
import javafx.scene.transform.*;
```

```
/**  
 * @author Amit  
 */
```

## Advanced Examples to JavaFX

```
var _Directoty_ = "http://192.168.10.13/images/userimages";
var pic="/13785home2.gif";
var hght=200;
var wdth=300;
SwingFrame {
  title: "Picture Show"
  width: 300
  height:290
  closeAction: function() {
    System.exit( 0 );
  }
  visible: true
  content: BorderPanel{
    center: Label {icon: bind Icon{image: Image {
      url:"{_Directoty_}{pic}"
      height:hght
      width:wdth
    }}
  }
}
bottom: GridPanel{
  rows:2
  columns:2
  content:[ Button {
    icon:Icon { image: Image {url:
      "{_Directoty_}{pic}"
      height:50
      width:50}}
    action: function() {
      pic="/13785home2.gif";
    }
  }, Button {
    icon:Icon { image: Image {
      url: "{_Directoty_}/2156mediaplayer.gif"
      height:50
      width:50}}
    action: function() {
      pic="/2156mediaplayer.gif";
    }
  }, Button {
    icon:Icon { image: Image {url:
      "{_Directoty_}/1532articleimagesf1.jpg"
      height:50
      width:50}}
    action: function() {
      pic="/1532articleimagesf1.jpg";
    }
  }, Button {
    icon:Icon { image: Image {url:
      "{__DIR__}/../resources/posit_1.gif"
      height:50
      width:50}}
  }
```

## Advanced Examples to JavaFX

```
        action: function() {
            hght+=10;
            width+=10;
        }
    }, Button {
        icon:Icon { image: Image {url:
            "{__DIR__}/../resources/nega_1.gif"
            height:50
            width:50}}
        action: function() {
            hght-=10;
            width-=10;
        }
    },Button {
        text:"Reset"
        height:50
        width:50
        action: function() {
            hght=200;
            width=300;
        }
    }
}
}
```

**Output:**



## Advanced Examples to JavaFX

We can “Zoom In” and Zoom Out” and “Reset” our images.



### Download Source Project

#### **(X) Creating Menu Bar and Menu Items in JavaFX Script**

To create Menu Bar and Menu Items we have to use “javafx.ext.swing” package. In this package we have “Menu” class and “Menu Item” class.

```
package javafxmenu;

import javafx.ext.swing.Menu;
import javafx.ext.swing.MenuItem;
import javafx.ext.swing.SwingFrame;
import javafx.ext.swing.Canvas;
import javafx.scene.paint.Color;
import javafx.ext.swing.BorderPanel;
import javafx.ext.swing.Label;

/**
 * @author Amit
 */

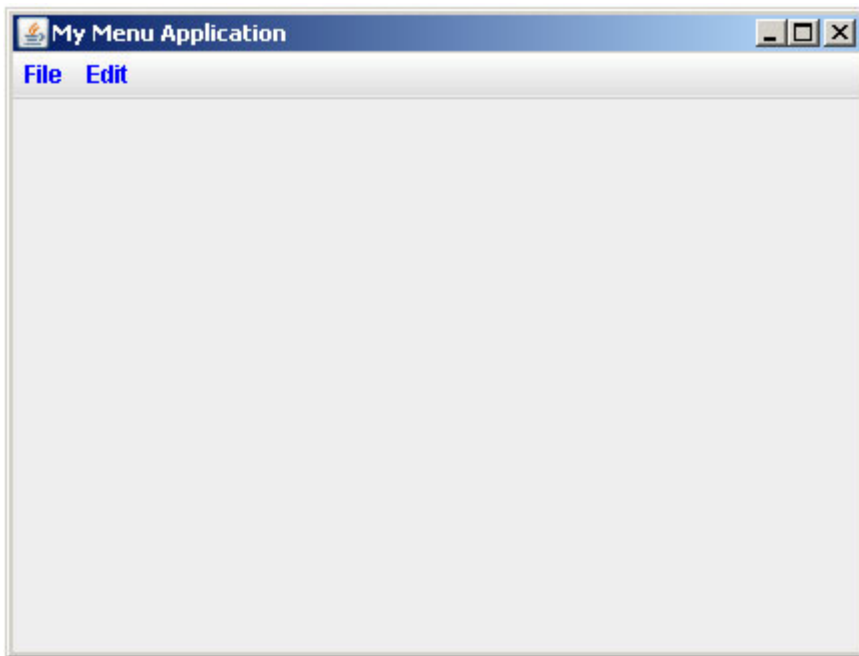
var newFrame = SwingFrame {
    title: "My New Frame"
    width: 500
    height: 500
    closeAction: function() {
        java.lang.System.exit( 0 );
    }
}
```

## Advanced Examples to JavaFX

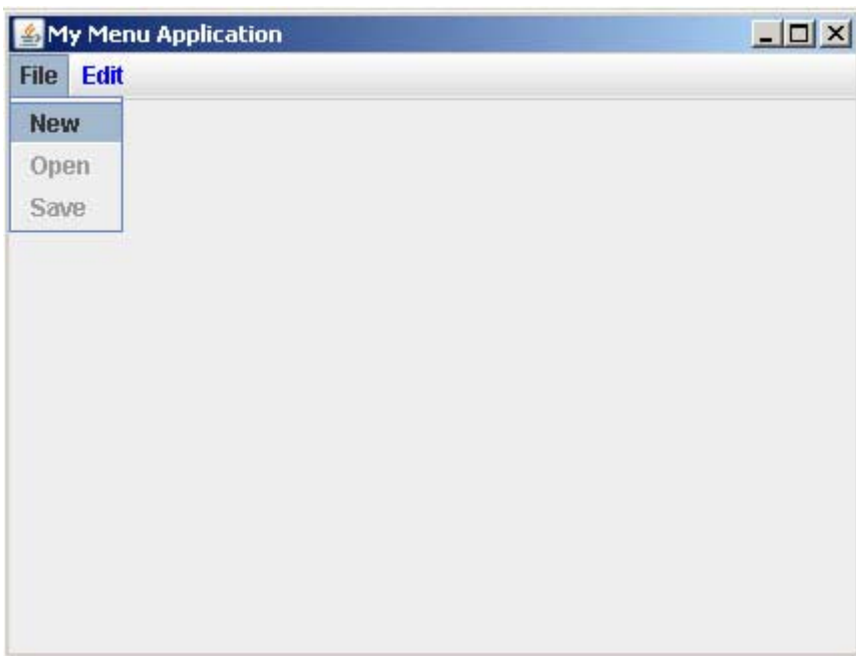
```
visible: false
content: BorderPanel {
    top: Label {
        text: "Welcome Amit Kumar !"
        foreground:Color.PINK
    }
}
}
SwingFrame {
    title: "My Menu Application"
    width: 700
    height: 600
    closeAction: function() {
        java.lang.System.exit( 0 );
    }
    visible: true
    menus: [ Menu {
        text: "File"
        items : [
            MenuItem {
                text: "New"
                action : function(){
                    newFrame.visible=true;
                }
            },MenuItem {
                text: "Open"
                enabled:false
            },MenuItem {
                text: "Save"
                enabled:false
            }
        ]
        foreground:Color.BLUE
    },Menu {
        text: "Edit"
        items : [
            MenuItem {
                text: "Undo"
            },MenuItem {
                text: "Redo"
            },MenuItem {
                text: "Find"
            },MenuItem {
                text: "Replace"
            }
        ]
        foreground:Color.BLUE
    } ]
}
```

# Advanced Examples to JavaFX

Output:



These menu items "File" and "Edit" also have its sub items.



**[Download Source Project](#)**

# Advertise with JavaJazzUp

We are the top most providers of technology stuffs to the java community. Our technology portal network is providing standard tutorials, articles, news and reviews on the Java technologies to the industrial technocrats. Our network is getting around 3 million hits per month and its increasing with a great pace.

For a long time we have endeavored to provide quality information to our readers. Furthermore, we have succeeded in the dissemination of the information on technical and scientific facets of IT community providing an added value and returns to the readers.

We have serious folks that depend on our site for real solutions to development problems.

**JavaJazzUp Network** comprises of :

<http://www.roseindia.net>  
<http://www.newstrackindia.com>  
<http://www.javajazzup.com>  
<http://www.allcooljobs.com>

## Advertisement Options:

Banner	Size	Page Views	Monthly
Top Banner	470*80	5,00,000	USD 2,000
Box Banner	125 * 125	5,00,000	USD 800
Banner	460x60	5,00,000	USD 1,200
Pay Links		Un Limited	USD 1,000
Pop Up Banners		Un Limited	USD 4,000

The <http://www.roseindia.net> network is the "real deal" for technical Java professionals. Contact me today to discuss your customized sponsorship program. You may also ask about advertising on other Technology Network.

Deepak Kumar  
[deepak@roseindia.net](mailto:deepak@roseindia.net)

## India's Cheapest web Service Provider

### Web Packages

#### Package (in INR)

Package Name	Price
Starter Package	Rs 5,100 + Taxes Extra
Business Package	Rs 9,111 + Taxes Extra
Corporate Package	Rs 22,500 + Taxes Extra
Smart Package	Rs 45,500 + Taxes Extra

\* Domain Registration free with every package

### Packages Specifications

#### Starter Package

- 5 Web Pages
- 10 Stock Images
- 5 POP 3 Accounts
- 50 MB Hosting Space
- Unlimited Edits

#### Business Package

- 10 Web Page Design
- Flash Site Animation
- 25 Stock Images
- 10 POP3 Accounts
- 50 MB Hosting Space
- Unlimited Edits

#### Corporate Package

- 25 Web Page Design
- Flash Site Animation
- Flash Intro Page
- 50 Stock Images
- 25 POP3 Accounts
- 100 MB Hosting Space
- Unlimited Edit

#### Smart Package

- 200 Web Page Design
- Catalog with 200 items
- Flash Site Animation
- Flash Intro Page
- Unlimited Stock Images
- 50 POP3 Accounts
- 250 MB Hosting Space
- Unlimited Edits

 **RoseIndia**  
Technologies Pvt. Ltd.

Logon to: <http://www.roseindia.net/services/>

# Valued JavaJazzup Readers Community

**We invite you to post Java-technology oriented stuff. It would be our pleasure to give space to your posts in JavaJazzup.**

## **Contribute to Readers Forum**

If theres something youre curious about, were confident that your curiosity, combined with the knowledge of other participants, will be enough to generate a useful and exciting Readers Forum. If theres a topic you feel needs to be discussed at JavaJazzup, its up to you to get it discussed.

## **Convene a discussion on a specific subject**

If you have a topic youd like to talk about . Whether its something you think lots of people will be interested in, or a narrow topic only a few people may care about, your article will attract people interested in talking about it at the Readers Forum. If you like, you can prepare a really a good article to explain what youre interested to tell java technocrats about.

## **Sharing Expertise on Java Technologies**

If youre a great expert on a subject in java, the years you spent developing that expertise and want to share it with others. If theres something youre an expert on that you think other technocrats might like to know about, wed love to set you up in the Readers Forum and let people ask you questions.

## **Show your innovation**

We invite people to demonstrate innovative ideas and projects. These can be online or technology-related innovations that would bring you a great appreciations and recognition among the java technocrats around the globe.

## **Hands-on technology demonstrations**

Some people are Internet experts. Some are barely familiar with the web. If youd like to show others aroud some familiar sites and tools, that would be great. It would be our pleasure to give you a chance to provide your demonstrations on such issues : How to set

up a blog, how to get your images onto Flickr, How to get your videos onto YouTube, demonstrations of P2P software, a tour of MySpace, a tour of Second Life (or let us know if there are other tools or technologies you think people should know about...).

## **Present a question, problem, or puzzle**

Were inviting people from lots of different worlds. We do not expect everybody at Readers Forum to be an expert in some areas. Your expertise is a real resource you may contribute to the Java Jazzup. We want your curiosity to be a resource, too. You can also present a question, problem, or puzzle that revolves around java technologies along with their solution that you think would get really appreciated by the java readers around the globe.

## **Post resourceful URLs**

If you think you know such URL links which can really help the readers to explore their java skills. Even you can post general URLs that you think would be really appreciated by the readers community.

## **Anything else**

If you have another idea for something youd like to do, talk to us. If you want to do something that we havent thought of, have a crazy idea, wed really love to hear about it. Were open to all sorts of suggestions, especially if they promote readers participation.



***WAKE UP YOUR PROGRAMMING KNOWLEDGE WITH  
ROSEINDIA.NET***





**Broadcasters are storytellers, newspapers are fact providers and we serve the both.**



<http://www.newstrackindia.com>