Getting Started with Infragistics Android Controls

Topic Overview

Purpose

This document provides detailed information on how to get started with Community Technology Preview of Infragistics Android controls. In addition, it provides instructions on how to create sample projects using the Infragistics Android controls.

In this topic

This topic contains the following sections:

- *Overview*
  - Introduction
  - Requirements
  - Infragistics Controls
  - Infragistics Samples
- *Project Setup*
  - Creating Android Project in Eclipse
  - Creating Android Project in IntelliJ
- Creating a Data Chart Using the Infragistics Android Controls
- Creating a Pie Chart Using the Infragistics Android Controls
- Creating a Radial Gauge Using the Infragistics Android Controls
- Creating a Linear Gauge Using the Infragistics Android Controls
- Creating a Bullet Graph Using the Infragistics Android Controls
- Running the Samples Browser on an Android device

Overview

Introduction

The Infragistics Android product allows you build native Android apps and visualize your data in over 50 different types of charts, gauges, bullet graph, and grid controls.

Requirements

In order to use Infragistics Android controls you need to meet the following requirements:

Software	Notes
ADP Eclipse or IntelliJ	Integrated Development Environment (IDE), which support development of Android applications
Infragistics Library Jar Files	The chart control requires references to Infragistics Libraries: • infragistics.core.jar • infragistics.datachart.jar • infragistics.dv.shared.jar • infragistics.ui.jar

Infragistics Controls

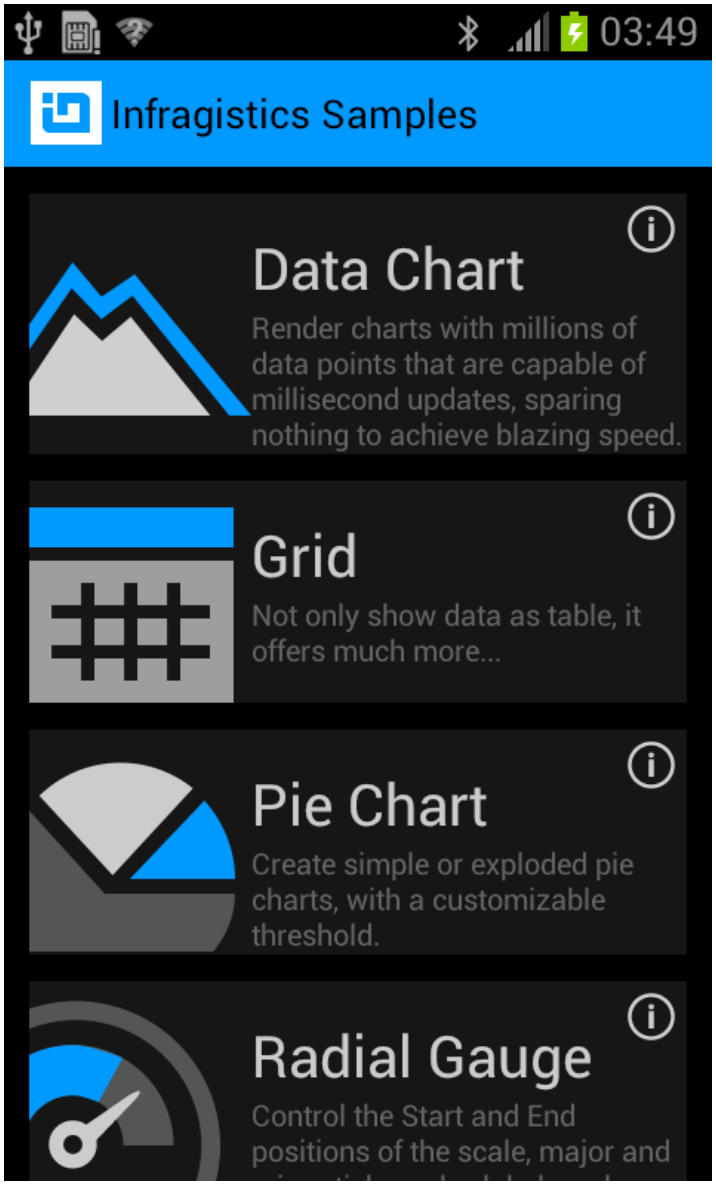
The Infragistics Android product provide the following controls:

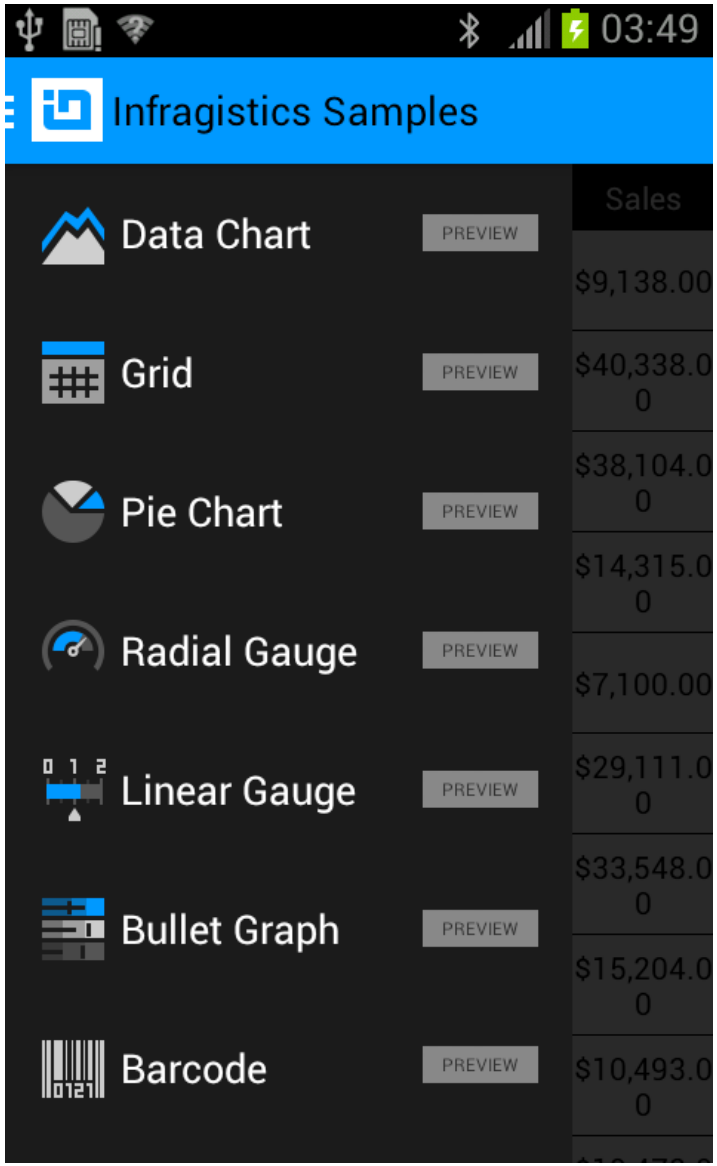
Control Name	Description
Data Chart View	Represents interactive charting control, including common series like Category, Stacked, Polar, Radial and Scatter, complete with trend lines, financial indicators, interactions, and animations.
Pie Chart View	Represents interactive pie chart control with customizable labels and locations of pie slices.
Grid View	Represents fully virtualized, data-bound, high performance data grid with pin-able columns, carousel views, selection events & customizable layouts.
Bullet Graph View	Represents a simple and concise view of key performance indicators (KPI) to dashboard users that presents progress towards goals and good/better/best ranges in as little horizontal or vertical space as possible.
Linear Gauge v	Represents highly configurable linear gauge that is easy to use to shows off KPIs with rich style and interactivity in dashboards.
Radial Gauge View	Represents highly configurable radial gauge that is easy to use to shows off KPIs with rich style and interactivity in dashboards.
Barcode View	Represents barcode control for displaying Code128 and QR codes.

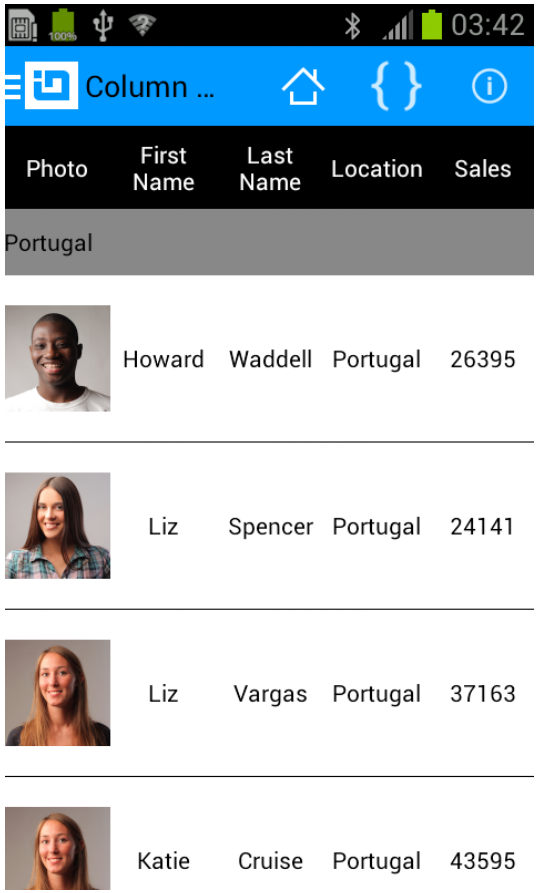
The CTP zip file contains Java Docs zip file and you can use it to gain better understanding of the APIs of Infragistics Android controls.

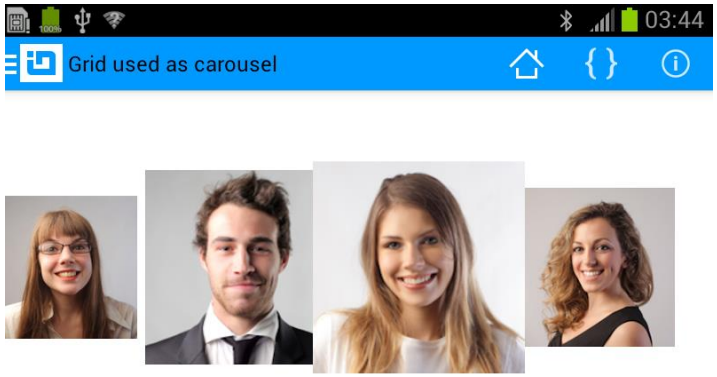
Infragistics Samples

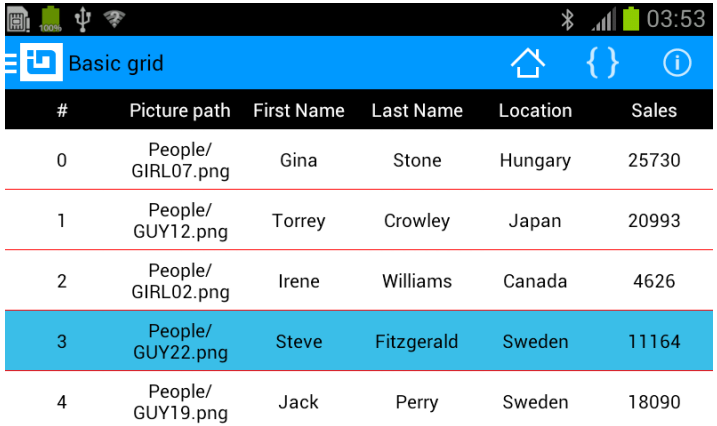
This document is part of the Community Technology Preview, which includes zipped samples browser APK file with samples displaying the data chart, radial and linear gauges, bullet graph, and grid controls. These samples are designed to help you to get up and running quickly with the Infragistics Android controls. In addition, you will find control specific samples within the CTP zip file. These are designed to help you to get up and running quickly with the Infragistics Android controls. Unzip these samples, save, and import it into your workspace. You can import this by selecting File -> Import -> Existing Android Code into Workspace, then select Next and Browse to the saved samples, and select OK. Then add the jars to the lib folders as described in next section of this document. Once that is done, build the samples and run on your device or emulator.


03:49
Infragistics Samples
Data Chart
Render charts with millions of data points that are capable of millisecond updates, sparing nothing to achieve blazing speed.
Grid
Not only show data as table, it offers much more...
Pie Chart
Create simple or exploded pie charts, with a customizable threshold.
Radial Gauge
Control the Start and End positions of the scale, major and

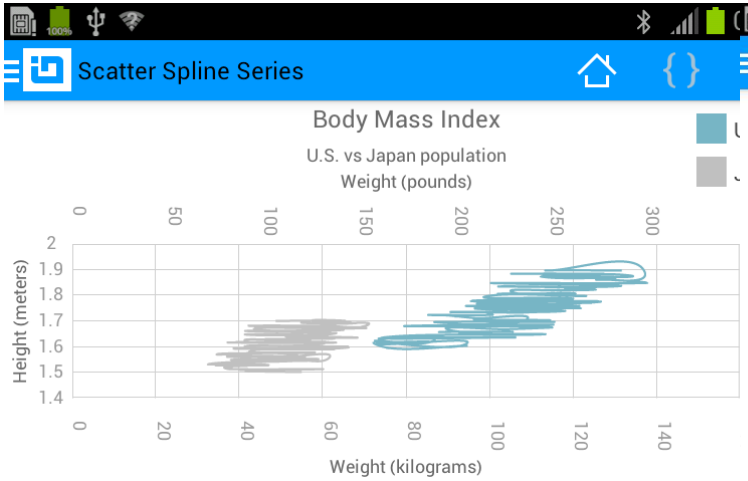

03:49
Infragistics Samples
Data Chart
PREVIEW
Grid
PREVIEW
Pie Chart
PREVIEW
Radial Gauge
PREVIEW
Linear Gauge
PREVIEW
Bullet Graph
PREVIEW
Barcode
PREVIEW
Sales
$9,138.00
$40,338.00
$38,104.00
$14,315.00
$7,100.00
$29,111.00
$33,548.00
$15,204.00
$10,493.00

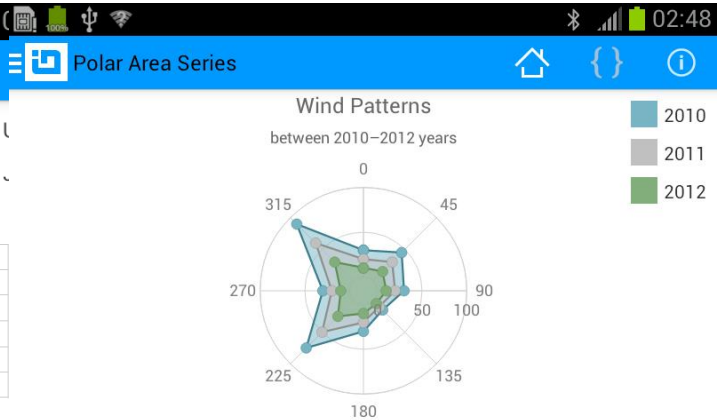






#	Picture path	First Name	Last Name	Location	Sales
0	People/ GIRL07.png	Gina	Stone	Hungary	25730
1	People/ GUY12.png	Torrey	Crowley	Japan	20993
2	People/ GIRL02.png	Irene	Williams	Canada	4626
3	People/ GUY22.png	Steve	Fitzgerald	Sweden	11164
4	People/ GUY19.png	Jack	Perry	Sweden	18090



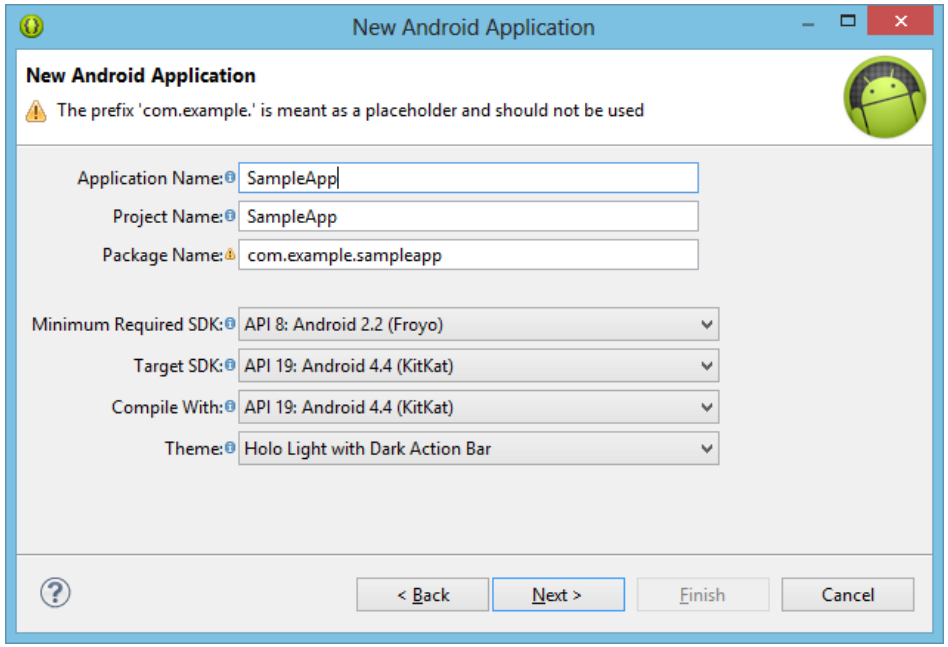


Project Setup

Creating Android Project in Eclipse

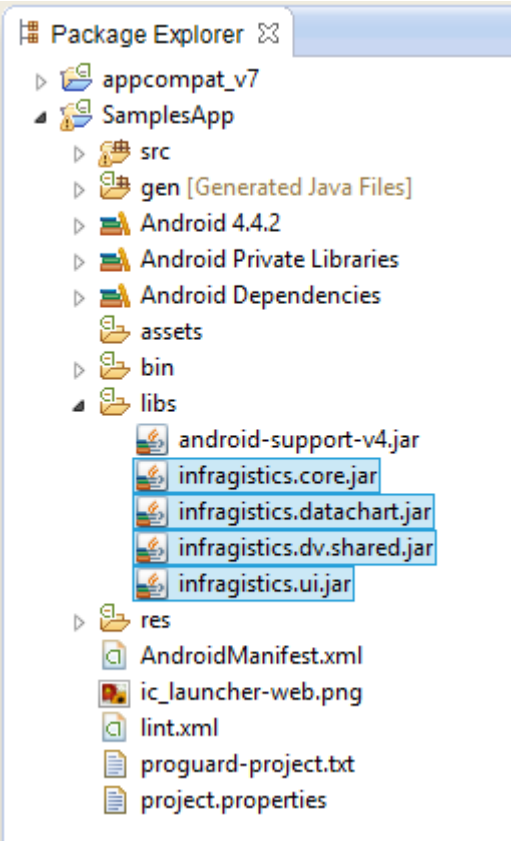
The following steps provide instructions on how to create Android project and reference Infragistics libraries in ADT Eclipse.

1. Open ADT Eclipse application
2. Create a new projects using Android Application preset.
3. Provide application name, e.g. SampleApp
4. Provide project name, e.g. SampleApp



5. Select all the defaults from the options
6. Add all required Infragistics library files:

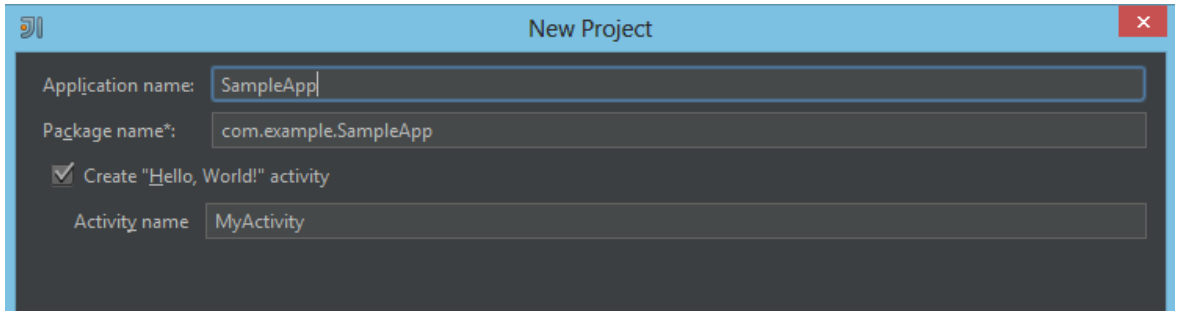
    - infragistics.core
    - infragistics.datachart
    - infragistics.dvshared
    - infragistics.gauges
    - infragistics.ui


Package Explorer
appcompat_v7
SamplesApp
src
gen [Generated Java Files]
Android 4.4.2
Android Private Libraries
Android Dependencies
assets
bin
libs
android-support-v4.jar
infragistics.core.jar
infragistics.datachart.jar
infragistics.dv.shared.jar
infragistics.ui.jar
res
AndroidManifest.xml
ic_launcher-web.png
lint.xml
proguard-project.txt
project.properties

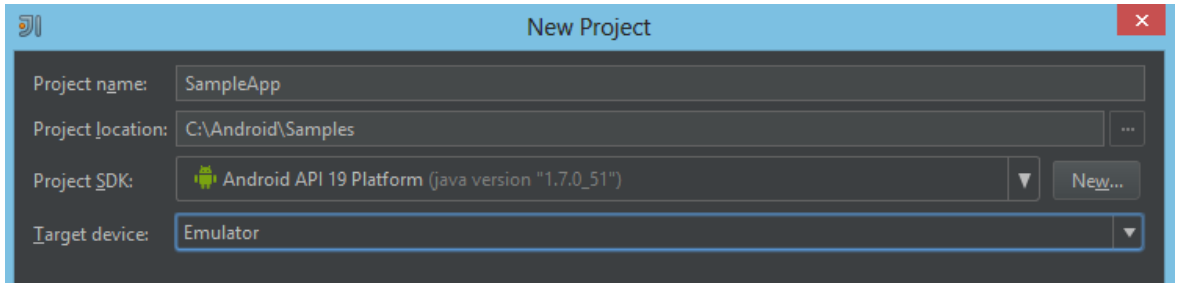
Creating Android Project in IntelliJ

The following steps demonstrate how to create Android project and reference Infragistics Android libraries in IntelliJ.

1. Open IntelliJ application
2. Create a new projects using Application Module from the Android presets.
3. Provide application name, e.g. SampleApp

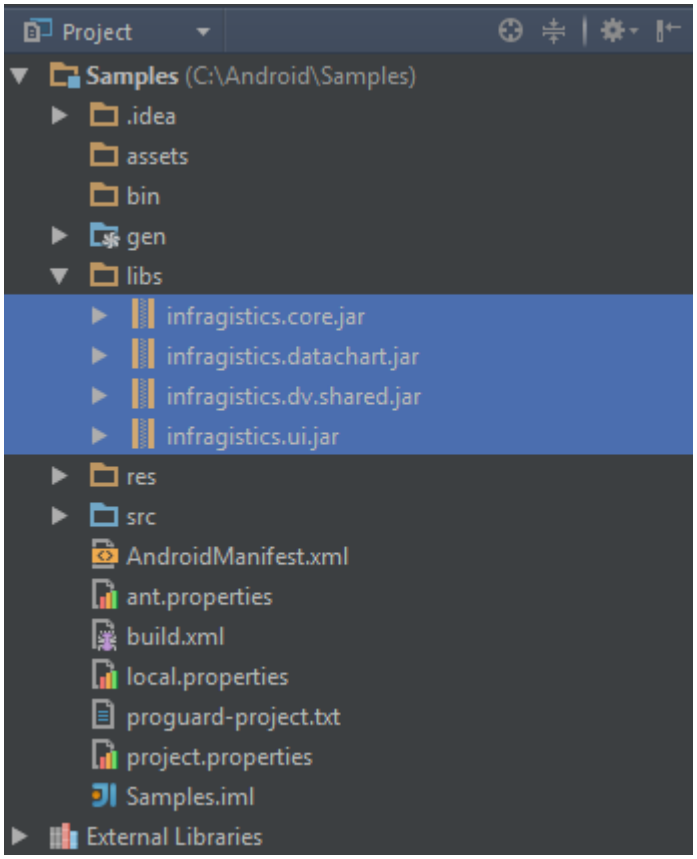


4. Provide project name, e.g. SampleApp
5. Select all the defaults from the options
6. Select Emulator as the target device



7. Add all required Infragistics library files:

- infragistics.core
- infragistics.datachart
- infragistics.dvshared
- infragistics.gauges
- infragistics.ui



Creating a Data Chart Using the Infragistics Android Controls

Setting up project

This section provides instruction for adding Infragistics Data Chart control to an Android project.

1. Follow instruction in the Project Setup section to create Android project and call it: **TestChartApp**

Adding data model

2. Add the following sample data file to the project:

Source File: TestDataItem.java

```
package com.example.gettingstarted;

public class TestDataItem {
	private double _value;
	public double getValue() {
		return _value;
	}
	public double setValue(double value) {
		_value = value;
		return value;
	}
	private String _label;
	public String getLabel() {
		return _label;
	}
	public String setLabel(String value) {
		_label = value;
		return value;
	}
}
```

Source File: TestData.java

```
package com.example.gettingstarted;

import java.util.ArrayList;
```

```java
public class TestData
            extends ArrayList<TestDataItem> {
      public TestData() {
            double curr = 10.0;
            double curr2 = 20.0;

            for (int i = 0; i < 31; i++) {
                  curr += Math.random() * 4.0 - 2.0;
                  curr2 += Math.random() * 4.0 - 2.0;

                  TestDataItem newItem = new TestDataItem();
                  newItem.setLabel(((Integer)(i + 1)).toString());
                  newItem.setValue(Math.min(curr, curr2));
                  add(newItem);
            }
      }
}
```

Adding Chart and binding data

3. Open MainActivity.java file and enter the following code:

Source File: MainActivity.java

```
import com.infragistics.controls.LineSeries;
import com.infragistics.controls.CategoryXAxis;
import com.infragistics.controls.DataChartView;
import com.infragistics.controls.NumericYAxis;

@Override
public View onCreateView(LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
      View rootView = inflater.inflate(R.layout.fragment_main, container, false);

      TestData data = new TestData();

      CategoryXAxis xAxis = new CategoryXAxis();
      NumericYAxis yAxis = new NumericYAxis();
      xAxis.setDataSource(data);
      xAxis.setLabel("Label");

      LineSeries lineSeries = new LineSeries();
      lineSeries.setXAxis(xAxis);
      lineSeries.setYAxis(yAxis);
      lineSeries.setValueMemberPath("Value");
      lineSeries.setDataSource(data);

      DataChartView chart = new DataChartView(rootView.getContext());
      chart.setHorizontalZoomable(true);
      chart.setVerticalZoomable(true);
      chart.addAxis(xAxis);
      chart.addAxis(yAxis);
      chart.addSeries(lineSeries);

      return(chart);
}
```

4. Build your project and run on your device / emulator

Creating a Pie Chart Using the Infragistics Android Controls

Setting up project

This section provides instruction for adding Infragistics Pie Chart control to an Android project.

1. Follow instruction in the Project Setup section to create Android project and call it: **TestPieChartApp**

Adding data model

2. Add the following sample data file to the project:

Source File: TestPieDataItem.java

```
package com.example.gettingstarted;

public class PieTestDataItem {
	private double _value;
	public double getValue() {
		return _value;
	}
	public double setValue(double value) {
		_value = value;
		return value;
	}
	public double highValue;
	public double lowValue;
	public String label;
}
```

Source File: TestPieData.java

```
package com.example.gettingstarted;

import java.util.ArrayList;

import com.infragistics.system.collections.generic.List__1;

public class PieTestData
	extends ArrayList<PieTestDataItem> {
```

```
    public PieTestData() {

        String[] dep = new String[] { "Admininstration", "Sales", "IT",
        "Marketing", "Development", "Support", "HR" };
        for (int i = 0; i < 7; i++) {
            PieTestDataItem newItem = new PieTestDataItem();
            newItem.label = dep[i];
            switch(i)
            {
                case 0 : newItem.highValue = 20 ; break ;
                case 1 : newItem.highValue = 40 ; break ;
                case 2 : newItem.highValue = 50 ; break ;
                case 3 : newItem.highValue = 40 ; break ;
                case 4 : newItem.highValue = 15 ; break ;
                case 5 : newItem.highValue = 20 ; break ;
                case 6 : newItem.highValue = 10 ; break ;
            }
            add(newItem);
        }
    }
}
```

Adding Pie Chart and binding data

3. Open MainActivity.java file and enter the following code:

Source File: MainActivity.java

```
import com.infragistics.controls.PieChartView;

@Override
public View onCreateView(LayoutInflater inflater,
                ViewGroup container, Bundle savedInstanceState) {
      View rootView = inflater.inflate(R.layout.fragment_main, container, false);


      PieTestData pdata = new PieTestData();
      PieChartView piechart = new PieChartView(rootView.getContext());
      piechart.setBackgroundColor(Color.WHITE);
      piechart.setDataSource(pdata);
      piechart.setLabelMemberPath("label");
      piechart.setValueMemberPath("highValue");
      piechart.setAllowSliceExplosion(true);
      piechart.setRadiusFactor(0.7);

      return(piechart);

}
```

4. Build your project and run on your device / emulator

Creating a Radial Gauge Using the Infragistics Android Controls

Setting up project

This section provides instruction for adding Infragistics Radial Gauge control to an Android project.

1. Follow instruction in the Project Setup section to create Android project and call it: **TestRadialGaugeApp**

Adding Radial Gauge

2. Open MainActivity.java file and enter the following code:

Source File: MainActivity.java

```
import com.infragistics.controls.SolidColorBrush;
import com.infragistics.controls.RadialGaugeRange;
import com.infragistics.controls.RadialGaugeView;

@Override
public View onCreateView(LayoutInflater inflater,
            ViewGroup container, Bundle savedInstanceState) {
      View rootView = inflater.inflate(R.layout.fragment_main,
                                    container, false);

      SolidColorBrush brush = new SolidColorBrush() ;
      brush.setColor(Color.parseColor("#C62D36"));
      RadialGaugeRange range1 = new RadialGaugeRange();
      range1.setStartValue(70);
      range1.setEndValue(100) ;
      range1.setBrush(brush) ;
      range1.setOuterStartExtent(0.45);
      range1.setOuterEndExtent(0.40);
      RadialGaugeView gauge = new RadialGaugeView(rootView.getContext());
      gauge.setInterval(20);
      gauge.setValue(40);
      gauge.setMaximumValue(120);
      gauge.setMinimumValue(20);
      gauge.addRange(range1);

      return(gauge);
}
```

3. Build your project and run on your device / emulator

Creating a Linear Gauge Using the Infragistics Android Controls

Setting up project

This section provides instruction for adding Infragistics Linear Gauge control to an Android project.

1. Follow instruction in the Project Setup section to create Android project and call it: **TestLinearGaugeApp**

Adding Radial Gauge

2. Open MainActivity.java file and enter the following code:

Source File: MainActivity.java

```
import com.infragistics.controls.SolidColorBrush;
import com.infragistics.controls.LinearGaugeView;
import com.infragistics.controls.LinearGraphRange;

@Override
public View onCreateView(LayoutInflater inflater,
            ViewGroup container, Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_main,
                                     container, false);

    SolidColorBrush brush = new SolidColorBrush() ;
    brush.setColor(Color.parseColor("#C62D36"));

    LinearGraphRange range1 = new LinearGraphRange();
    range1.setStartValue(70);
    range1.setEndValue(100) ;
    range1.setBrush(brush);

    LinearGaugeView gauge = new LinearGaugeView(rootView.getContext());
    int height = (int)TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP,
                   200, getResources().getDisplayMetrics());
    RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(
              RelativeLayout.LayoutParams.MATCH_PARENT, height);
              gauge.setLayoutParams(params);
    gauge.setTransitionDuration(1000);
    gauge.setInterval(20);
    gauge.setValue(40);
    gauge.setMaximumValue(120);
    gauge.setMinimumValue(20);
    gauge.addRange(range1);

    rootView.removeAllViews();
    rootView.addView(gauge);

    return rootView;
}
```

3. Build your project and run on your device / emulator

Creating a Bullet Graph Using the Infragistics Android Controls

Setting up project

This section provides instruction for adding Infragistics Bullet Graph control to an Android project.

1. Follow instruction in the Project Setup section to create Android project and call it: **TestBulletGraphGaugeApp**

Adding Radial Gauge

2. Open MainActivity.java file and enter the following code:

Source File: MainActivity.java

```
import com.infragistics.controls.SolidColorBrush;
import com.infragistics.controls.LinearGraphRange;
import com.infragistics.controls.BulletGraphView;

@Override
public View onCreateView(LayoutInflater inflater,
            ViewGroup container, Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_main,
                                     container, false);

    BulletGraphView gauge = new BulletGraphView(rootView.getContext());

    int height = (int)TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP,
                200, getResources().getDisplayMetrics());
    RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(
             RelativeLayout.LayoutParams.MATCH_PARENT, height);
    gauge.setLayoutParams(params);

    gauge.setTransitionDuration(1000);
    gauge.setInterval(20);
    gauge.setValue(40);
    gauge.setMaximumValue(120);
    gauge.setMinimumValue(0);

    SolidColorBrush b = new SolidColorBrush();
    b.setColor(Color.WHITE);
    SolidColorBrush brush = new SolidColorBrush();
    brush.setColor(Color.LTGRAY);
    SolidColorBrush brush1 = new SolidColorBrush();
    brush1.setColor(Color.GRAY);
    SolidColorBrush brush2 = new SolidColorBrush();
    brush2.setColor(Color.DKGRAY);

    LinearGraphRange lgrange1 = new LinearGraphRange();
    lgrange1.setStartValue(0);
    lgrange1.setEndValue(20);
    lgrange1.setBrush(brush);

    LinearGraphRange lgrange2 = new LinearGraphRange();
    lgrange2.setStartValue(20);
    lgrange2.setEndValue(60) ;
    lgrange2.setBrush(brush1) ;
```

```
    LinearGraphRange lgrange3 = new LinearGraphRange();
    lgrange3.setStartValue(60);
    lgrange3.setEndValue(120) ;
    lgrange3.setBrush(brush2) ;

    gauge.addRange(lgrange1);
    gauge.addRange(lgrange2);
    gauge.addRange(lgrange3);
    gauge.setValueBrush(b);
    gauge.setTargetValueBrush(b);

    rootView.removeAllViews();
    rootView.addView(gauge);

    return rootView;
}
```

3. Build your project and run on your device / emulator

Running the Samples Browser on an Android device

Instructions

This section provides instruction on how to run the sample browser on a device using provided APK.

1. Browse to the location of Android SDK, navigate to Platforms-Tools folder
2. SHIFT+RIGHT CLICK and select Open Command Window
3. ADB Install LOCATION_OF_APK\SampleBrowser.apk

Note, it is best to just have the selected device installed, otherwise you will have to find the device's ID to install the APK on to