B4X Booklets

B4A B4i B4J

B4X XUI

B4**X** **U**ser **I**nterface

Edition 2.5

Last update: 2024.01.05

Main contributors: Klaus Christl (klaus), Erel Uziel (Erel)

To search for a given word or sentence use the Search function in the Edit menu.

All the source code and files needed (layouts, images etc.) of the example projects in this booklet are included in the SourceCode folder.

Updated for:
B4A version 12.80
B4i version 8.50
B4J version 10.00

B4X Booklets:
B4X Getting Started
B4X Language
B4X IDE Integrated Development Environment
B4X Visual Designer
B4X Help tools

B4XPages Cross-platform projects
B4X CustomViews
B4X Graphics
B4X XUI B4X User Interface
B4X SQLite Database
B4X JavaObject NativeObject

B4R Example Projects

You can consult these booklets online in this link [B4X] Documentation Booklets.
Be aware that external links don’t work in the online display.

1 B4X platforms

B4X is a suite of programming languages for different platforms.

B4X suite supports more platforms than any other tool
ANDROID | IOS | WINDOWS | MAC | LINUX | ARDUINO | RASPBERRY PI | ESP8266 | AND MORE...

- **B4A** **Android**

  B4A is a **100% free** development tool for Android applications, it includes all the features needed to quickly develop any type of Android app.

- **B4i** **iOS**

  B4i is a development tool for native iOS applications.
  B4i follows the same concepts as B4A, allowing you to reuse most of the code and build apps for both Android and iOS.

- **B4J** **Java / Windows / Mac / Linux / Raspberry PI**

  B4J is a **100% free** development tool for desktop, server and IoT solutions.
  With B4J you can easily create desktop applications (UI), console programs (non-UI) and server solutions.
  The compiled apps can run on Windows, Mac, Linux and ARM boards (such as Raspberry Pi).

- **B4R** **Arduino / ESP8266**

  B4R is a **100% free** development tool for native Arduino and ESP8266 programs.
  B4R follows the same concepts of the other B4X tools, providing a simple and powerful development tool.
  B4R, B4A, B4J and B4i together make the best development solution for the Internet of Things (IoT).

- **B4XPages**

  B4XPages is an internal library for B4A, B4i and B4J allowing to develop easily cross-platform programs.
  B4XPages is explained in detail in the B4XPages Cross-platform projects booklet.
  Even, if you want to develop only in one platform it is interesting to use the B4XPages library it makes the program flow simpler especially for B4A.

2 General

XUI (B4**X** **U**ser **I**nterface) are Cross platform & native UI libraries.

The purpose of XUI library is to make it easier to share code between B4A, B4J and B4i projects.
It is an important new library and Erel expects that all B4X developers who target more than a single platform will use it at some point.

Two main concepts:
- The XUI libraries API (application programming interface) is the same between all three libraries.
- It is very simple to switch between the XUI objects and the native objects when needed.

The second point is important. XUI types provide a different "view" or a different wrapper above the same native objects. They do not replace the native types.
There are some new features provided in the libraries which you can use even if you target a single platform.

3 XUI Objects

The XUI objects below are supported.

- B4XView
  Any view or node can be assigned to a B4XView object.
  The B4XView type includes all the common methods, including methods that are not available in all types.

- B4XBitmap
  Wrapper for Bitmap (B4A, B4i) and Image (B4J).
  Adds all the features available in B4A and B4i Bitmap type to B4J
  (Resize, Crop, Rotate and saving as Jpeg).

- B4XCanvas
  Cross platform canvas.

- B4X Font
  Wrapper for the TypeFace object (B4A) and the Font object (B4i, B4J). Used for graphics.

- B4XPath
  Wraps:
  B4A android.graphics.Path,
  B4i UIBezierPath,
  B4J com.sun.javafx.geom.Path2D.

- B4XRect
  Wraps B4ARect, B4iRect and B4JRect. Used for graphics.

- XUI
  The XUI object includes various methods and utilities.

3.1 B4XView

Any view or node can be assigned to a B4XView object.
The B4XView type includes all the common methods, including methods that are not available in all types.

Members:

AddView (View As android.view.View, Left As Int, Top As Int, Width As Int, Height As Int)
Adds a view.
Supported types
B4A - Activity, Panel
B4i - Panel
B4J - Pane

BringToFront
Changes the Z order of this view and brings it to the front.

Checked As Boolean
Gets or sets the checked state (also named selected or value).
Supported types:
B4A - CheckBox, RadioButton, ToggleButton, Switch
B4i - Switch
B4J: CheckBox, RadioButton, ToggleButton.

Color As Int
Gets or sets the background color. Returns 0 if the color is not available.

EditTextHint As String
Gets or sets the hint or prompt text. Supported types:
B4A - EditText
B4i - TextField
B4J - TextArea, TextField

Enabled As Boolean
Gets or sets whether the view is enabled. Does nothing if the view does not support this property.

Font As B4XFont
Gets or sets the font (typeface and text size).
Supported types:
B4A - EditText, Label, Button, CheckBox, RadioButton, ToggleButton
B4i - TextField, TextView, Button, Label
B4J - Sets the Font property if available. Otherwise sets the CSS attribute.

GetAllViewsRecursive As IterableList
Returns an iterator that iterates over all the child views including views that were added to other child views.
Make sure to check the view type as it might return subviews as well.
Example:

```
For Each v As B4XView In Panel1.GetAllViewsRecursive
   ...
Next
```

Supported types
B4A - Activity, Panel
B4i - Panel
B4J - Pane

GetView (Index As Int) As B4XView
Returns the B4XView view at the given index.
Supported types
B4A - Activity, Panel
B4i - Panel
B4J - Pane

Height As Int
Gets or sets the view's height.

IsInitialized As Boolean

Left As Int
Gets or sets the left position.

LoadLayout (LayoutFile As String)
Loads the layout file.
Supported types
B4A - Panel
B4i - Panel
B4J - Pane

NumberOfViews As Int [read only]
Returns the number of direct child views.
Supported types
B4A - Activity, Panel
B4i - Panel
B4J - Pane

Parent As B4XView [read only]
Returns the parent. The object returned will be uninitialized if there is no parent.

Progress As Int
Gets or sets the progress value. The progress value is scaled between 0 to 100 (this is different than the native views range in B4J and B4i).
Supported types:
B4A - ProgressBar
B4J - ProgressView, ProgressIndicator
B4i - ProgressView
Value should be between 0 to 100.

RemoveAllViews
Removes all views.
Supported types
B4A - Activity, Panel
B4i - Panel
B4J - Pane

RemoveViewFromParent
Removes the view from its parent.

RequestFocus As Boolean
Requests focus to be set to this view. Returns True if the focus has shifted.
Always returns True in B4J.

Rotation As Float
Gets or sets the view's rotation transformation (in degrees).

ScrollViewContentHeight As Int
Gets or set the scroll view inner panel height.
Supported types:
B4A - HorizontalScrollView, ScrollView
B4i - ScrollView
B4J - ScrollPane

ScrollViewContentWidth As Int
Gets or set the scroll view inner panel width.
Supported types:
B4A - HorizontalScrollView, ScrollView
B4i - ScrollView
B4J - ScrollPane

ScrollViewInnerPanel As B4XView [read only]
Gets or sets the scroll view inner panel.
Supported types:
B4A - HorizontalScrollView, ScrollView
B4i - ScrollView
B4J - ScrollPane

ScrollViewOffsetX As Int
Gets or sets the horizontal scroll position.
Supported types:
B4A - HorizontalScrollView (returns 0 for ScrollView).
B4i - ScrollView
B4J - ScrollPane

ScrollViewOffsetY As Int
Gets or sets the vertical scroll position.
Supported types:
B4A - ScrollView (returns 0 for HorizontalScrollView).
B4i - ScrollView
B4J - ScrollPane

SendToBack
Changes the Z order of this view and sends it to the back.

SetBitmap (Bitmap As Bitmap)
Sets the view's bitmap.
Supported types:
B4A - All views. The view's Drawable will be set to a BitmapDrawable with Gravity set to CENTER.
B4i - ImageView. ContentMode set to Fit.
B4J - ImageView. PreserveRatio is set to True.

SetColorAndBorder (BackgroundColor As Int, BorderWidth As Int, BorderColor As Int, BorderCornerRadius As Int)
Sets the background color and border.
B4A - The view's drawable will be set to ColorDrawable.

SetColorAnimated (Duration As Int, FromColor As Int, ToColor As Int)
Changes the background color with a transition animation between the FromColor and the ToColor colors.
Duration - Animation duration measured in milliseconds.

SetLayoutAnimated (Duration As Int, Left As Int, Top As Int, Width As Int, Height As Int)
Sets the view size and position.
Duration - Animation duration in milliseconds. Pass 0 to make the change immediately.

SetRotationAnimated (Duration As Int, Degree As Float)
Rotates the view with animation.
Duration - Animation duration in milliseconds.
Degree - Rotation degree.

SetTextAlignment (Vertical As String, Horizontal As String)
Sets the text alignment.
Vertical - TOP, CENTER or BOTTOM.
Horizontal - LEFT, CENTER or RIGHT.

In B4i the vertical alignment has no effect.
Supported types:
B4A - EditText, Label, Button, CheckBox, RadioButton, ToggleButton
B4J - Label, Button, Checkbox, RadioButton, ToggleButton

SetTextSizeAnimated (Duration As Int, TextSize As Double)
Sets the text size.
Supported types:
B4A - EditText, Label, Button, CheckBox, RadioButton, ToggleButton
B4i - TextField, TextView, Button, Label. Only labels are animated.
B4J - Sets the TextSize property if available and the CSS attribute for other types.

SetVisibleAnimated (Duration As Int, Visible As Boolean)
Fades in or fades out the view.

Snapshot As B4XBitmap
Captures the views appearance.

Tag As Object
Gets or sets the view's tag object.

Text As String
Gets or sets the text. Supported types:
B4A - EditText, Label, Button, CheckBox, RadioButton, ToggleButton
B4i - TextField, TextView, Button, Label
B4J - TextField, TextArea, Label, Button, CheckBox, RadioButton, ToggleButton

TextColor As Int
Gets or sets the text colors. Supported types:
B4A - EditText, Label, Button, CheckBox, RadioButton, ToggleButton
B4i - TextField, TextView, Label
B4J - All types. Based on the native TextColor property if available or -fx-text-fill CSS attribute.

TextSize As Float
Gets or sets the text size.
Supported types:
B4A - EditText, Label, Button, CheckBox, RadioButton, ToggleButton
B4i - TextField, TextView, Button, Label
B4J - Returns the TextSize property if available and the CSS attribute for other types. Returns 0 if attribute not available.

Top As Int
Gets or sets the top position.

TOUCH_ACTION_DOWN As Int
Constant for the Touch event.

TOUCH_ACTION_MOVE As Int
Constant for the Touch event.

TOUCH_ACTION_ MOVE_NOTOUCH As Int
Constant for the Touch event.
Equivalent to MouseMoved in B4J (will never be raised in B4A or B4i).

TOUCH_ACTION_UP As Int
Constant for the Touch event.

Visible As Boolean
Gets or sets whether the view is visible.

Width As Int
Gets or sets the view's width.

3.2 B4XBitmap

Represents a loaded image.

Similar to:

- **B4A** Bitmap
- **B4i** Bitmap
- **B4J** Image

Members:

Crop (Left As Int, Top As Int, Width As Int, Height As Int) As B4XBitmap
Returns a **new** cropped bitmap.

Height As Double [read only]
Returns the bitmap's height.

IsInitialized As Boolean
There is no **Initialize**, you can use `xui.LoadBitmap` to load an image.

Resize (Width As Int, Height As Int, KeepAspectRatio As Boolean) As B4XBitmap
Returns a **new** bitmap with the given width and height.

Rotate (Degrees As Int) As B4XBitmap
Returns a **new** rotated bitmap. The bitmap will be rotated clockwise.
The following values are supported on all three platforms: 0, 90, 180, 270.

Scale As Float
Returns the bitmap scale. It will always be 1 in B4J and B4i.

Width As Double [read only]
Returns the bitmap's width.

WriteToStream (Out As java.io.OutputStream, Quality As Int, Format As android.graphics.Bitmap.CompressFormat)
Writes the bitmap to the output stream.
Quality - Value between 0 (smaller size, lower quality) to 100 (larger size, higher quality), which is a hint for the compressor for the required quality.
Format - JPEG or PNG.

3.2.1 Examples

To use the xui methods you need to declare the library in the Globals routine like:

```
  Private xui As XUI
```

3.2.1.1 Define a B4XBitmap

Declare a B4xBitmap.

```
  Private xbmpBackground As B4XBitmap
```

3.2.1.2 Initialize the Bimap (B4A, B4i) or Image (B4J)

```
'initialize the background bitmap
  xbmpBackground = xui.LoadBitmap(File.DirAssets, "rose.jpg")
```

3.2.1.3 Draw the bitmap

csvBackground is a B4XCanvas.

```
  'draw the background image
  cvsBackground.DrawBitmap(xbmpBackground, xbckRect)
```

3.2.1.4 Save a bitmap

```
  Dim Out As OutputStream
#If B4A
  Out = File.OpenOutput(File.DirRootExternal, "Test.png", False)
#Else If B4i
  Out = File.OpenOutput(File.DirDocuments, "Test.png", False)
#Else If B4J
  Out = File.OpenOutput(File.DirApp, "Test.png", False)
#End If
  cvsLayer(2).CreateBitmap.WriteToStream(Out, 100, "PNG")
  Out.Close
```

B4i, File.DirDocuments can be shared through iTunes.
In order to enable this feature you need to add this attribute:

```
#PlistExtra: <key>UIFileSharingEnabled</key><true/>
```

Or with the `xui.SetDataFolder` and the `xui.DefaultFolder` method.

```
  'saves the bitmap of layer(2)
  Dim Out As OutputStream
  xui.SetDataFolder("SimpleDrawMethods")
  Out = File.OpenOutput(xui.DefaultFolder, "Test.png", False)
  cvsLayer(2).CreateBitmap.WriteToStream(Out, 100, "PNG")
  Out.Close
```

`xui.DefaultFolder`

- B4A - Same as File.DirInternal.
- B4i - Same as File.DirDocuments.
- B4J - Same as File.DirData. You must first call SetDataFolder once before you can use this folder.

3.3 B4XCanvas

A cross platform canvas.

Members:

ClearRect (Rect As B4XRect)
Clears the given rectangle. Does not work in B4J with clipped paths.

ClipPath (Path As B4XPath)
Clips the drawings to a closed path.

CreateBitmap As B4XBitmap
Returns a copy of the canvas bitmap. In B4A it returns the canvas bitmap itself (not a copy).

DrawBitmap (Bitmap As android.graphics.Bitmap, Destination As B4XRect)
Draws a bitmap in the given destination. Use B4XBitmap.Crop to draw part of a bitmap.

DrawBitmapRotated (Bitmap As android.graphics.Bitmap, Destination As B4XRect, Degrees As Float)
Similar to DrawBitmap. Draws a rotated bitmap.

DrawCircle (x As Float, y As Float, Radius As Float, Color As Int, Filled As Boolean, StrokeWidth As Float)
Draws a circle.

DrawLine (x1 As Float, y1 As Float, x2 As Float, y2 As Float, Color As Int, StrokeWidth As Float)
Draws a line between x1,y1 to x2,y2.

DrawPath (Path As B4XPath, Color As Int, Filled As Boolean, StrokeWidth As Float)
Draws a path.
Draws the given path.
Path - Path shape.
Color - Drawing color.
Filled - Whether to fill the shape or not.
StrokeWidth - Stroke width. Only relevant when Filled is False.

Note that there is a subtle difference in the way the stroke width affects the drawing between B4J and the other platforms.
In B4J the path defines the stroke edge. In B4A and B4i it defines the stroke center.

DrawPathRotated (Path As B4XPath, Color As Int, Filled As Boolean, StrokeWidth As Float, Degrees As Float, CenterX As Float, CenterY As Float)
Similar to DrawPath. Rotates the path based on the degrees and center parameters.

DrawRect (Rect As B4XRect, Color As Int, Filled As Boolean, StrokeWidth As Float)
Draws a rectangle.

DrawText (Text As String, x As Float, y As Float, Font As B4XFont, Color As Int, Alignment As android.graphics.Paint.Align)
Draws the text.
Text - The text that will be drawn.
x - The origin X coordinate.
y - The origin Y coordinate.
Font - The text font.
Color - Drawing color.
Alignment - Sets the alignment relative to the origin. One of the following values: LEFT, CENTER, RIGHT.

DrawTextRotated (Text As String, x As Float, y As Float, Font As B4XFont, Color As Int, Alignment As android.graphics.Paint.Align, Degree As Float)
Similar to DrawText. Rotates the text before it is drawn.

Initialize (View As B4XView)
Initializes the canvas.
In B4A and B4i the canvas will draw on the passed view.
In B4J the canvas which is a view by itself is added to the passed pane as the first element.

Invalidate
Commits the drawings. Must be called for the drawings to be updated.

MeasureText (Text As String, Font As B4XFont)
Measures single line texts and returns their width, height and the height above the baseline.
Rect.Top returns the height above the baseline.
Code to draw center aligned text:

```
Dim r As B4XRect = cvs1.MeasureText(Text, Fnt)
Dim BaseLine As Int = CenterY - r.Height / 2 - r.Top
cvs1.DrawText(Text, CenterX, BaseLine, Fnt, Clr, "CENTER")
```

Returns : B4XRect

Release
Releases native resources related to the canvas. Does nothing in B4A and B4J.

RemoveClip
Removes a previously set clip region.

Resize (Width As Float, Height As Float)
Resizes the canvas.

TargetRect As B4XRect [read only]
Returns a B4XRect with the same dimensions as the target view.

TargetView As B4XView [read only]
Returns the target view.

3.3.1 Examples

We use dip values for all coordinates for compatibility with B4A.
The dip function does nothing in B4i nor in B4J.

The source code can be found in the GraphicSourceCode\SimpleDrawMethods

To use the xui methods you need to declare the library in the Globals routine like:

```
Private xui As XUI
```

3.3.1.1 Declare a target B4XView

```
Private xplMain As B4XView
```

3.3.1.2 Initialize a B4XCanvas

```
Private cvsBackground, cvsLayer(3) As B4XCanvas
```

3.3.1.2.1 B4A

```
xplMain = Activity      'Target view
cvsBackground.Initialize(xplMain)
```

3.3.1.2.2 B4i

```
xplMain = Page1.RootPanel    'Target view
cvsBackground.Initialize(xplMain)
```

3.3.1.2.3 B4J

```
xplMain = MainForm.RootPane   'Target view
cvsBackground.Initialize(xplMain)
```

3.3.1.3 Draw a line

With variables instead of numeric values.

```
x1 = 10dip
y1 = 10dip
x2 = 250dip
y2 = 40dip
cvsLayer(2).DrawLine(x1, y1, x2, y2, xui.Color_RGB(0, 128, 255), 8dip)
```

3.3.1.4 Draw a rectangle

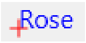
```
' draw the rectangles on layer(0)
xRect.Initialize(40dip, 330dip, 150dip, 120dip)
cvsLayer(0).DrawRect(xRect, xui.Color_ARGB(128, 0, 0, 255), True, 1)
```

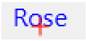
3.3.1.5 Draw a circle

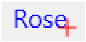
```
'draw an empty circle
xc = 180dip
xc = 120dip
Radius = 50dip
cvsLayer(2).DrawCircle(xc, yc, Radius, xui.Color_Red, False, 5dip)
```

3.3.1.6 Draw a text

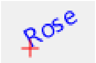
```
' draw a horizontal text
x1 = 350dip
y1 = 40dip
cvsLayer(2).DrawText("Rose", x1, y1, xFont, xui.Color_Red, "LEFT")
y1 = 70dip
cvsLayer(2).DrawText("Rose", x1, y1, xFont, xui.Color_Red, "CENTER")
y1 = 100dip
cvsLayer(2).DrawText("Rose", x1, y1, xFont, xui.Color_Red, "RIGHT")
```







```
' draw a rotated text
x1 = 500dip
y1 = 40dip
cvsLayer(2).DrawTextRotated("Rose", x1, y1, xFont, xui.Color_Red, "LEFT", -30)
```

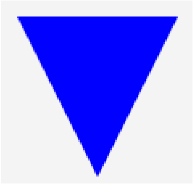


See also the MeasureText method.

3.3.1.7 ClipPath

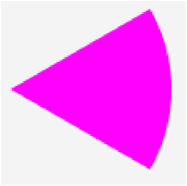
`cvsLayer` is a B4XCanvas.

The code below draws a clipped triangle.



```
'draw a clipped triangle
mPath.Initialize(400dip, 400dip)
mPath.LineTo(500dip, 400dip)
mPath.LineTo(450dip, 500dip)
cvsLayer(2).ClipPath(mPath)
cvsLayer(2).DrawRect(cvsLayer(2).TargetRect, xui.Color_Blue, True, 1dip)
cvsLayer(2).RemoveClip
```

The code below draws a clipped arc.



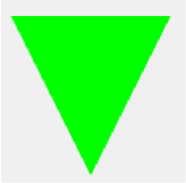
```
'draw a clipped arc
Private mPath As B4XPath
xc = 250dip
yc = 400dip
Radius = 100dip
mPath.InitializeArc(xc, yc, Radius, -30, 60)
cvsLayer(2).ClipPath(mPath)
cvsLayer(2).DrawRect(cvsLayer(2).TargetRect, xui.Color_Magenta, True, 1dip)
cvsLayer(2).RemoveClip
```

3.3.1.8 DrawPath

`cvsLayer` is a B4XCanvas.

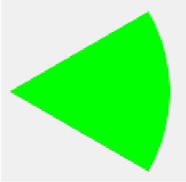
The code below draws a triangle path.

```
'draw a triangle path
Private mPath As B4XPath
mPath.Initialize(170dip, 550dip)
mPath.LineTo(270dip, 550dip)
mPath.LineTo(220dip, 650dip)
cvsLayer(2).DrawPath(mPath, xui.Color_Green, True, 1dip)
cvsLayer(2).RemoveClip
```

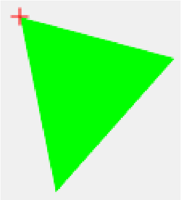


The code below draws an arc path.

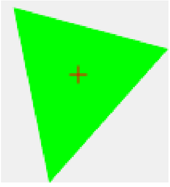
```
'draw an arc Path
Private mPath As B4XPath
xc = 50dip
yc = 600dip
Radius = 100dip
mPath.InitializeArc(xc, yc, Radius, -30, 60)
cvsLayer(2).DrawPath(mPath, xui.Color_Green, True, 1dip)
```



The codes below draw a rotated triangle path with two different rotation centers.



```
'draw a rotated triangle path with the center
Private mPath As B4XPath
mPath.Initialize(300dip, 550dip)
mPath.LineTo(400dip, 550dip)
mPath.LineTo(350dip, 650dip)
cvsLayer(2).DrawPathRotated(mPath, xui.Color_Green, True, 1dip, 15, 300dip, 550dip)
cvsLayer(2).RemoveClip
DrawCross(300dip, 550dip, xui.Color_Yellow)
```



```
'draw a rotated triangle path with the center
Private mPath As B4XPath
mPath.Initialize(400dip, 550dip)
mPath.LineTo(500dip, 550dip)
mPath.LineTo(450dip, 650dip)
cvsLayer(2).DrawPathRotated(mPath, xui.Color_Green, True, 1dip, 15, 450dip, 580dip)
cvsLayer(2).RemoveClip
DrawCross(450dip, 580dip, xui.Color_Red)
```

3.3.1.9 DrawPolygon

This is not a ‘standard’ method, but a XUI code routine proposed by Erel in the forum.
It uses DrawPath for B4A and B4i but uses the jFx fillPolygon or strokePolygon method for B4J with JavaObject.

```
Sub DrawPolygon (cvs1 As B4XCanvas, Points As List, Color As Int, Filled As Boolean, St
rokeWidth As Double)
   If Points.Size < 1 Then Return

#If B4A or B4i
   Dim FirstPoint() As Int = Points.Get(0)
   Dim p As B4XPath
   p.Initialize(FirstPoint(0),  = Points.Get(i)
       p.LineTo(point(0), point(1))
   Next
   cvs1.DrawPath(p, Color, Filled, StrokeWidth)
#Else
   Dim jcvs As JavaObject = cvs1
   jcvs = jcvs.GetFieldJO("cvs").RunMethodJO("getObject", Null).RunMethod("getGraphicsC
ontext2D", Null)
   jcvs.RunMethod("save", Null)
   Dim xpoints(Points.Size), ypoints(Points.Size) As Double
   For i = 0 To Points.Size - 1
       Dim point() As Int = Points.Get(i)
       xpoints(i) = point(0)
       ypoints(i) = point(1)
   Next
   Dim paint As Object = fx.Colors.From32Bit(Color)
   If Filled Then
       jcvs.RunMethod("setFill", Array(paint))
       jcvs.RunMethod("fillPolygon", Array(xpoints, ypoints, Points.Size))
   Else
       jcvs.RunMethod("setStroke", Array(paint))
       jcvs.RunMethod("setLineWidth", Array(StrokeWidth))
       jcvs.RunMethod("strokePolygon", Array(xpoints, ypoints, Points.Size))
   End If
   jcvs.RunMethod("restore", Null)
#End If
   cvs.Invalidate
End Sub
```

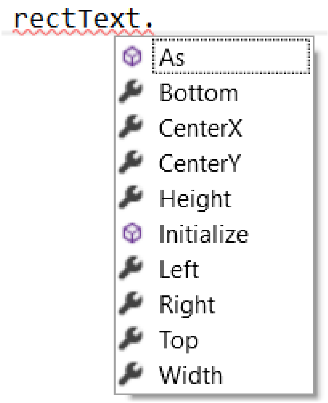
And an example:

```
  DrawPolygon(cvs, Array (Array As Int(100dip, 100dip), Array As Int(200dip, 0), _
       Array As Int (300dip, 100dip), Array As Int(200dip, 200dip), Array As Int(100dip
, 100dip)), xui.Color_Red, True, 2dip)
```

3.3.1.10 MeasureText

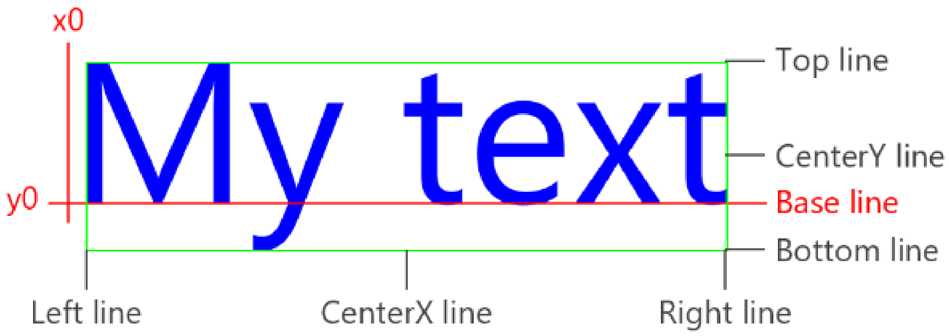
When you draw text there are situation where you need to know the dimensions of the text.
This can be done with the MeasureText method, which returns a B4XRect rectangle object.
In the sketch below, x0 and y0 are the coordinates given in the DrawText routine.
Y0 is the base line of the text.

Meaning of the rectangle properties:



Bottom = distance of the base line to the Bottom
CenterX = distance of the base line to the CenterX
CenterY = distance of the base line to the CenterY
Height = Bottom - Top

Left = distance of the base line to the Left
Right = distance of the base line to the Right
Top = distance of the base line to the Top
Width = Right – Left



- Left line = x0 + rectText.Left
- CenterX line = x0 + rectText.CenterX
- Right line = x0 + rectText.Right
- Width = rectText.Right - rectText.Left
- Top line = y0 + rectText.Top
- CenterY line = y0 + rectText.CenterY
- Bottom line = y0 + rectText.Bottom
- Height = rectText.Bottom - rectText.Top

```
   Private pnlTest As B4XView
   Private cvsTest As B4XCanvas

   Private fntText As B4XFont
   Private rectText As B4XRect          'the text rectangle
   Private rectTextOuter As B4XRect     'the outer text rectangle
   fntText = xui.CreateDefaultFont(100)
   Text = "My text"
   rectText = cvsTest.MeasureText(Text, fntText)
   rectTextOuter.Initialize(x0 + rectText.Left, y0 + rectText.Top, x0 + rectText.Right, 
y0 + rectText.Bottom)
   cvsTest.DrawText(Text, x0, y0, fntText, xui.Color_Blue, "LEFT")
```

3.4 B4XFont

B4XFont is a wrapper for the objects below:

- B4A TypeFace object.
- B4i Font object.
- B4J Font object.

Members:

IsInitialized As Boolean [read only]

Size As Float [read only]

ToNativeFont As TypefaceWrapper
Returns a native font object representing the same font. A font object for B4i and B4J and a TypeFace object for B4A.

3.4.1 Examples

To use the xui methods you need to declare the library in the Globals routine like:

```
Private xui As XUI
```

3.4.1.1 Example with a 'standard' font

Declare a font directly, same code for all three products, `xLabelFont` is a B4XFont object:

```
Private xLabelFont As B4XFont

xLabelFont = xui.CreateDefaultFont(16)        ‘ default font

xLabelFont = xui.CreateDefaultBoldFont(16)    ‘ default bold font
```

3.4.1.2 Example with FontAwesome

```
Private xIconFont As B4XFont
#If B4A
   xIconFont = xui.CreateFont(Typeface.CreateNew(Typeface.FONTAWESOME,
Typeface.STYLE_NORMAL), 10)
#Else If B4i
   xIconFont = Font.CreateFontAwesome (10)
#Else If B4J
   xIconFont = fx.CreateFontAwesome(10)
#End If
xLabel.Font = xIconFont
```

3.4.1.3 Example with MaterialIcons

```
Private xIconFont As B4XFont
#If B4A
   xIconFont = xui.CreateFont(Typeface.CreateNew(Typeface.MATERIALICONS,
Typeface.STYLE_NORMAL), 10)
#Else If B4i
   xIconFont = xui.CreateFont(Font.CreateMaterialIcons(10))
#Else If B4J
   xIconFont = xui.CreateFont(fx.CreateFont("Material Icons", 10, False, False))
#End If
```

3.4.1.4 Example for a CustomView recovering the Label font

`xIconFont` is a B4XFont object.

```
Public Sub DesignerCreateView (Base As Object, Lbl As Label, Props As Map)
#If B4A
   xIconFont = xui.CreateFont(Lbl.Typeface, Lbl.TextSize)
#Else
   xIconFont = Lbl.Font
#End If
```

3.4.1.5 Example with a custom font

Sets a font with TimesNewRoman font style, `xfntTitle` is a B4XFont object.
Specific needs:

- B4A The Font file in the Assets folder, in this case *times.ttf*.
- B4i Nothing special.
- B4J Needs the JFX library.

```
Private xfntTitle As B4XFont
#If B4A
   Private tpf As Typeface
   tpf = Typeface.LoadFromAssets("times.ttf")
   xfntTitle = xui.CreateFont(tpf, 16)
#Else If B4i
   Private fnt As Font
   fnt = Font.CreateNew2("Times New Roman", 16)
   xfntTitle = xui.CreateFont(fnt, 16)
#Else If B4J
   Private fnt As Font
   fnt = fx.CreateFont("Times New Roman", 16, False, False)
   xfntTitle = xui.CreateFont(fnt, 16)
#End If
```

Or the short way:

```
Private xfntTitle As B4XFont
#If B4A
   xfntTitle = xui.CreateFont(Typeface.LoadFromAssets("times.ttf"), 16)
#Else If B4i
   xfntTitle = xui.CreateFont(Font.CreateNew2("Times New Roman", 16), 16)
#Else If B4J
   xfntTitle = xui.CreateFont(fx.CreateFont("Times New Roman", 16, False, False), 16)
#End If
```

3.5 B4XPath

B4XPath wraps the objects below:

- **B4A**: Process object - wraps android.graphics.Path
- **B4i**: Process object - wraps UIBezierPath
- **B4J**: Process object - wraps com.sun.javafx.geom.Path2D

Members:

Initialize (x As Float, y As Float) As B4XPath
Initializes the path and sets the value of the first point.

InitializeArc (x As Float, y As Float, Radius As Float, StartingAngle As Float, SweepAngle As Float) As B4XPath
Initializes the path and sets the current path shape to an arc.
x / y - Arc center.
Radius - Arc radius.
StartingAngle - The starting angle. 0 equals to hour 3.
SweepAngle - Sweep angle. Positive = clockwise.

InitializeOval (Rect As B4XRect) As B4XPath
Initializes the path and sets the current path shape to an oval.
Rect - The oval framing rectangle.

InitializeRoundedRect (Rect As B4XRect, CornersRadius As Float) As B4XPath
Initializes the path and sets the current path shape to a rectangle with rounded corners.
Rect - Rectangle.
CornersRadius - Corners radius.

IsInitialized As Boolean

LineTo (x As Float, y As Float) As B4XPath
Adds a line from the last point to the specified point.

You can use Paths to clip parts of a bitmap or draw a path, see examples ClipPath and DrawPath.

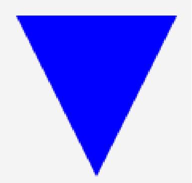
3.5.1 Examples

To use the xui methods you need to declare the library in the Globals routine like:

```
Private xui As XUI
```

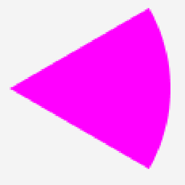
3.5.1.1 ClipPath

The code below draws a clipped triangle.



```
'draw a clipped triangle
mPath.Initialize(400dip, 400dip)
mPath.LineTo(500dip, 400dip)
mPath.LineTo(450dip, 500dip)
cvsLayer(2).ClipPath(mPath)
cvsLayer(2).DrawRect(cvsLayer(2).TargetRect, xui.Color_Blue, True, 1dip)
cvsLayer(2).RemoveClip
```

The code below draws a clipped arc.



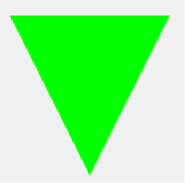
```
'draw a clipped arc
Private mPath As B4XPath
xc = 250dip
yc = 400dip
Radius = 100dip
mPath.InitializeArc(xc, yc, Radius, -30, 60)
cvsLayer(2).ClipPath(mPath)
cvsLayer(2).DrawRect(cvsLayer(2).TargetRect, xui.Color_Magenta, True, 1dip)
cvsLayer(2).RemoveClip
```

3.5.1.2 DrawPath

`cvsLayer` is a B4XCanvas.

The code below draws a triangle path.

```
'draw a triangle path
Private mPath As B4XPath
mPath.Initialize(170dip, 550dip)
mPath.LineTo(270dip, 550dip)
mPath.LineTo(220dip, 650dip)
cvsLayer(2).DrawPath(mPath, xui.Color_Green, True, 1dip)
cvsLayer(2).RemoveClip
```



3.6 B4XRect

Members:

Bottom As Float
Gets or sets the rectangle Bottom property.

CenterX As Float [read only]
Returns the horizontal center.

CenterY As Float [read only]
Returns the vertical center.

Height As Float
Gets or sets the rectangle height.

Initialize (Left As Float, Top As Float, Right As Float, Bottom As Float)
Initializes the rectangle with the given properties.

Left As Float
Gets or sets the rectangle Left property.

Right As Float
Gets or sets the rectangle Right property.

Top As Float
Gets or sets the rectangle Top property.

Width As Float
Gets or sets the rectangle width.

3.6.1 Examples

To use the xui methods you need to declare the library in the Globals routine like:

```
Private xui As XUI
```

3.6.1.1 Declare a rectangle

```
Private xRect As B4XRect
```

3.6.1.2 Initialize rectangle and draw it

```
' draw the rectangles on layer(0)
xRect.Initialize(40dip, 330dip, 150dip, 120dip)
cvsLayer(0).DrawRect(xRect, xui.Color_ARGB(128, 0, 0, 255), True, 1)
```

3.6.1.3 Set a rectangle to transparent ClearRect

```
' set the rectangle on layer(0) to transparent
xRect.Initialize(40dip, 330dip, 150dip, 120dip)
cvsLayer(0).ClearRect(xRect)
```

3.7 As keyword

The B4X objects are supersets of the original objdects.

The As keyword allows to switch between objects.

Example, setting the Padding for a B4A Label in the code, but the Label was declared as a B4XView.

Declaration of the B4XView:

```
    Private lblTest As B4XView
```

Instead of using the code below to convert the B4XView into the standard B4A view:

```
    Private lbl As Label
    lbl = lblTest
    lbl.Padding = Array As Int(5dip, 0, 5dip, 0)
```

Use this code:

```
    lblTest.As(Label).Padding = Array As Int(5dip, 0, 5dip, 0)
```

Another example with several Buttons and the same Click event routine.

Instead of:

```
Private Sub btnEvent_Click
    Private btnSender As Button

    btnSender = Sender

    Select btnSender.Tag
        Case "BS"
```

You can use:

```
Private Sub btnEvent_Click
    Select Sender.As(Button).Tag
        Case "BS"
```

4 Compatibilities B4A B4i B4J XUI

A list of current objects, which can be almost the same, or having different names with similar functionalities and / or the B4XView equivalent or having an equivalent CustomView.

B4J	B4A	B4i	XUI	CustomView
Button	Button	Button	B4XView	---
Canvas	Canvas	Canvas	B4XCanvas	---
CheckBox	CheckBox	Switch	---	B4XSwitch
ComboBox	Spinner	Picker	---	B4XComboBox
ImageView	ImageView	ImageView	B4XView	---
Image	Bitmap	Bitmap	B4XBitmap	---
Label	Label	Label	B4XView	---
ListView	ListView	---	---	xCustomListView
Pane	Panel	Panel	B4XView	
ProgressBar ProgressIndicator	ProgressBar	ProgressView	B4XView	---
ScrollPane	ScrollView HorizontalScrollView	ScrollView	B4XView	---
Slider	SeekBar	Slider	---	B4XSeekbar
TextField	EditText	TextField	B4XView	---
WebView	WebView	WebView	---	---

xCustomListView is a standard library.
B4XSwitch, B4XComboBox and B4XSeekbar are included in the XUI Views.b4xlib library, which is also a standard library.

5 XUI Process objects

To use the xui methods or objects you need to declare the library in the Globals routine like:

```
Private xui As XUI
```

Members:

Color_ARGB (Alpha As Int, R As Int, G As Int, B As Int) As Int
Returns the color value from the components. Values should be between 0 to 255.

Color_Black As Int

Color_Blue As Int

Color_Cyan As Int

Color_DarkGray As Int

Color_Gray As Int

Color_Green As Int

Color_LightGray As Int

Color_Magenta As Int

Color_Red As Int

Color_RGB (R As Int, G As Int, B As Int) As Int
Returns the color value from the components. Values should be between 0 to 255.

Color_Transparent As Int

Color_White As Int

Color_Yellow As Int

CreateDefaultBoldFont (Size As Float) As B4XFont
Create a new B4XFont with the default bold typeface.
Do NOT use DIP units with font sizes.

CreateDefaultFont (Size As Float) As B4XFont
Create a new B4XFont with the default typeface.
Do NOT use DIP units with font sizes.

CreateFont
Creates a new B4XFont from the given font and size.
Do NOT use DIP units with font sizes.

- **B4A**
  CreateFont (Typeface As android.graphics.Typeface, Size As Float) As B4XFont

- **B4i**
  CreateFont (Font.CreateNew2(Name As String, Size As Float) As B4XFont

- **B4J** needs the jFX library
  CreateFont (fx.CreateFont(FamilyName As String, Size As Float, Bold As Boolean, Italic As Boolean) As B4XFont

CreateFont2 (B4XFont As B4XFont, Size As Float) As B4XFont
Create a new B4XFont from the given B4XFont and size.
Do NOT use DIP units with font sizes.

CreateFontAwesome (Size As Float) As B4XFont
Creates a new B4XFont based on FontAwesome fonte.
Do NOT use DIP units with font sizes.

CreateMaterialIcons (Size As Float) As B4XFont
Creates a new B4XFont based on Material Icons font.
Do NOT use DIP units with font sizes.

CreatePanel (EventName As String) As B4XView
Creates a Panel (or Pane in B4J).
Note that the panel created will clip its child views.
In B4A, this method can only be called from an Activity context.

DefaultFolder As String [read only]
B4A - Same as File.DirInternal.
B4i - Same as File.DirDocuments.
B4J - Same as File.DirData. You must first call SetDataFolder once before you can use this folder.

DialogResponse_Cancel As Int

DialogResponse_Negative As Int

DialogResponse_Positive As Int

FileUri (Dir As String, FileName As String) As String
Returns a file uri. This can be used with WebView to access local resources.
The FileName parameter will be url encoded.
Example:

```
WebView1.LoadHtml($"<img src="${xui.FileUri(File.DirAssets, "smiley.png")}" />"$)
'or:
WebView1.LoadUrl($"${xui.FileUri(File.DirAssets, "smiley.png")}"$)
```

IsB4A As Boolean [read only]
Returns True in B4A.

IsB4i As Boolean [read only]
Returns True in B4i.

IsB4J As Boolean [read only]
Returns True in B4J.

LoadBitmap (Dir As String, FileName As String) As B4XBitmap
Loads a bitmap. In most cases you should use LoadBitmapResize instead.

LoadBitmapResize (Dir As String, FileName As String, Width As Int, Height As Int, KeepAspectRatio As Boolean) As B4XBitmap
Loads and resizes a bitmap.

Msgbox2Async (Message As CharSequence, Title As CharSequence, Positive As String, Cancel As String, Negative As String, Icon As BitmapWrapper) As Object
Shows a non-modal Msgbox.
Returns an object that can be used as the sender filter parameter for the Msgbox_Result event.
Message - Dialog message.
Title - Dialog title.
Positive - Positive button text. Pass an empty string to remove button.
Cancel - Cancel button text. Pass an empty string to remove button.
Negative - Negative button text. Pass an empty string to remove button.
Icon - Dialog icon. Pass Null to remove. Does nothing in B4i.
Example:

```
Dim sf As Object = xui.Msgbox2Async("Delete file?", "Title", "Yes", "Cancel", "No",
Null)
Wait For (sf) Msgbox_Result (Result As Int)
If Result = xui.DialogResponse_Positive Then
  Log("Deleted!!!")
End If
```

MsgboxAsync (Message As CharSequence, Title As CharSequence) As Object
Shows a non-modal Msgbox.
Returns an object that can be used as the sender filter parameter for the optional Msgbox_Result event.
Example:xui.MsgboxAsync("Hello", "World")

PaintOrColorToColor (Color As Object) As As Int
B4A, B4i - Does nothing.
B4J - Converts a Paint object to an Int color. Does not do anything if the color is already an Int color.

Scale As Float
Returns the screen normalized scale.
Always returns 1 in B4J and B4i.
Returns the same value as 100dip / 100 in B4A. Same as GetDeviceLayoutValues.Scale

SetDataFolder (AppName As String)
B4A, B4i - Does nothing.
B4J - Sets the subfolder name on Windows. The actual path will be similar to: C:\Users\[user name]\AppData\Roaming\[AppName].
Does nothing on other platforms.

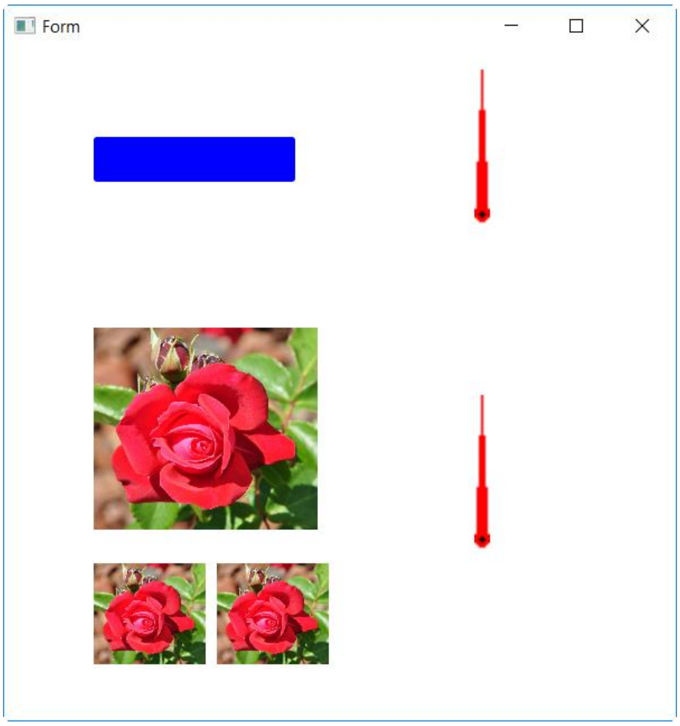
SubExists (Component As Object, Sub As String, NotUsed As Int) As Boolean
Same as SubExists keyword. Adds an additional parameter that is required in B4i (number of parameters).

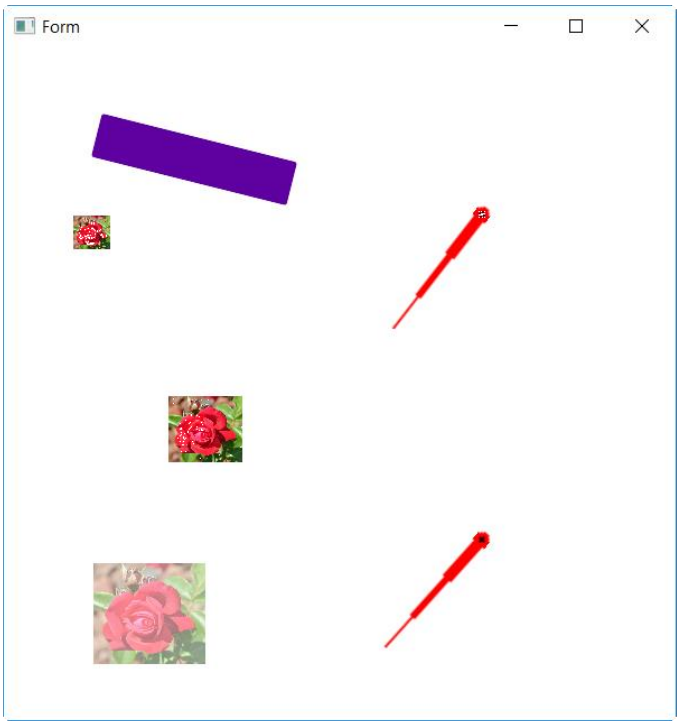
6 First example

This example project shows:

- 1 rotating panel with color changes.
- 1 continuous rotating needle, and 1 needle rotating by steps, like second needles.
- 1 picture disappearing and appearing.

The program uses a Timer for the changes.





The images show the B4J version.

6.1 The code

Most of the code is common to the three operating systems.
The variable declaration is a bit different in B4A Process_Globals and Globals.
The layout variable initialization is done in Activity_Create in B4A and in the Resize routines in B4i and B4J.

6.1.1 B4A

```
Sub Process_Globals
   Private xui As XUI
   Private tmrTest As Timer
End Sub

Sub Globals
   Private pnlTest As B4XView
   Private imvTest1, imvTest2, imvTest3 As B4XView
   Private imvNeedle1, ximvNeedle2 As B4XView

   Private pnlAngle = 0 As Double
   Private NeedleAngle = 0 As Double

   Private imvTest1Left, imvTest1Top, imvTest1Width, imvTest1Height As Double
   Private imvTest1CenterX, imvTest1CenterY As Double
   Private imvTest3Left, imvTest3Top, imvTest3Width, imvTest3Height As Double
End Sub

Sub Activity_Create(FirstTime As Boolean)
   Activity.LoadLayout("Main")

   InitObjects

   InitLayoutVariables
End Sub
```

6.1.2 B4i

The initialization of the layout variable for the ximvTest object are set in the Page1_Resize routine to make sure that the object dimensions are known.

```
Sub Process_Globals
   Public App As Application
   Public NavControl As NavigationController
   Private Page1 As Page

   Private xui As XUI
   Private tmrTest As Timer

   Private pnlTest As B4XView
   Private imvTest1, imvTest2, imvTest3 As B4XView
   Private imvNeedle1, ximvNeedle2 As B4XView

   Private pnlAngle = 0 As Double
   Private NeedleAngle = 0 As Double

   Private imvTest1Left, imvTest1Top, imvTest1Width, imvTest1Height As Double
   Private imvTest1CenterX, imvTest1CenterY As Double
   Private imvTest3Left, imvTest3Top, imvTest3Width, imvTest3Height As Double
End Sub

Private Sub Application_Start (Nav As NavigationController)
   'SetDebugAutoFlushLogs(True) 'Uncomment if program crashes before all logs are
printed.
   NavControl = Nav
   Page1.Initialize("Page1")
   Page1.Title = "Page 1"
   Page1.RootPanel.Color = Colors.White
   Page1.RootPanel.LoadLayout("Main")
   NavControl.ShowPage(Page1)

   InitObjects
End Sub

Private Sub Page1_Resize(Width As Int, Height As Int)
   InitLayoutVariables
End Sub
```

6.1.3 B4J

The initialization of the layout variable for the ximvTest object are set in the MainForm_Resize routine to make sure that the object dimensions are known.

```
Sub Process_Globals
   Private fx As JFX
   Private MainForm As Form
   Private xui As XUI
   Private tmrTest As Timer

   Private pnlTest As B4XView
   Private imvTest1, imvTest2, imvTest3 As B4XView
   Private imvNeedle1, ximvNeedle2 As B4XView

   Private pnlAngle = 0 As Double
   Private NeedleAngle = 0 As Double

   Private imvTest1Left, imvTest1Top, imvTest1Width, imvTest1Height As Double
   Private imvTest1CenterX, imvTest1CenterY As Double
   Private imvTest3Left, imvTest3Top, imvTest3Width, imvTest3Height As Double
End Sub

Sub AppStart (Form1 As Form, Args() As String)
   MainForm = Form1
   MainForm.RootPane.LoadLayout("Main") 'Load the layout file.
   MainForm.Show

   InitObjects
End Sub

'Return true to allow the default exceptions handler to handle the uncaught exception.
Sub Application_Error (Error As Exception, StackTrace As String) As Boolean
   Return True
End Sub

Private Sub MainForm_Resize (Width As Double, Height As Double)
   InitLayoutVariables
End Sub
```

6.1.4 Common code

The code below is exactly the same for all three projects.

```
Private Sub InitObjects
   'initialize and enable the timer
   tmrTest.Initialize("tmrTest", 1000)
   tmrTest.Enabled = True
End Sub

Private Sub InitLayoutVariables
   'define variables for the layout animation
   imvTest1Left = imvTest1.Left
   imvTest1Top = imvTest1.Top
   imvTest1Width = imvTest1.Width
   imvTest1Height = imvTest1.Height
   imvTest1CenterX = imvTest1Left + imvTest1Width / 2
   imvTest1CenterY = imvTest1Top + imvTest1Height / 2

   imvTest3Left = imvTest3.Left
   imvTest3Top = imvTest3.Top
   imvTest3Width = imvTest3.Width
   imvTest3Height = imvTest3.Height
End Sub

Private Sub tmrTest_Tick
   pnlAngle = pnlAngle + 45
   NeedleAngle = NeedleAngle + 6

   pnlTest.SetRotationAnimated(1000, pnlAngle)
   If pnlTest.Color = xui.Color_Blue Then
      pnlTest.SetColorAnimated(800, xui.Color_Blue, xui.Color_Red)
   Else
      pnlTest.SetColorAnimated(800, xui.Color_Red, xui.Color_Blue)
   End If
   imvNeedle1.SetRotationAnimated(1000, NeedleAngle)

   imvNeedle2.Rotation = NeedleAngle

   If imvTest1.Width < imvTest1Width Then
      imvTest1.SetLayoutAnimated(500, imvTest1Left, imvTest1Top, imvTest1Width, 
imvTest1Height)
   Else
      imvTest1.SetLayoutAnimated(500, imvTest1CenterX, imvTest1CenterY, 1dip, 1dip)
   End If

   If imvTest2.Visible = True Then
      imvTest2.SetVisibleAnimated(700, False)
   Else
      imvTest2.SetVisibleAnimated(700, True)
   End If

   If imvTest3.Width < imvTest3Width Then
      imvTest3.SetLayoutAnimated(500, imvTest3Left, imvTest3Top, imvTest3Width, 
imvTest3Height)
   Else
      imvTest3.SetLayoutAnimated(500, 0, 0, 1dip, 1dip)
   End If
End Sub
```

7 Graphics first steps

The XUI cross platform project, a B4XPages project, is in:
GraphicsSourceCode\GraphicsFirstSteps

I left platform specific projects in:
GraphicsSourceCode\GraphicsFirstStepsOld\B4A\
GraphicsSourceCode\GraphicsFirstStepsOld\B4i\
GraphicsSourceCode\GraphicsFirstStepsOld\B4J\

To draw something, we need a Canvas object which is simply a drawing tool.

XUI
B4XCanvas is a cross platform Canvas, it is a wrapper of the native Canvases of B4A, B4i and B4J.
It must be ‘connected’ to a B4XView object in the Initialize method.
The example project is a B4XPages project, which has as a CustomView with the SAME module for the three platforms.

B4A, B4i
The Canvas draws onto a Bitmap. This Bitmap can be the background bitmap of views.
The most common views to draw on are: Activity, Panel, ImageView or a Bitmap.

The Canvas must be 'connected' to a bitmap or a view background image in the Initialize method.

- Initialize(Target View)
- Initialize2(Target Bitmap) B4A only

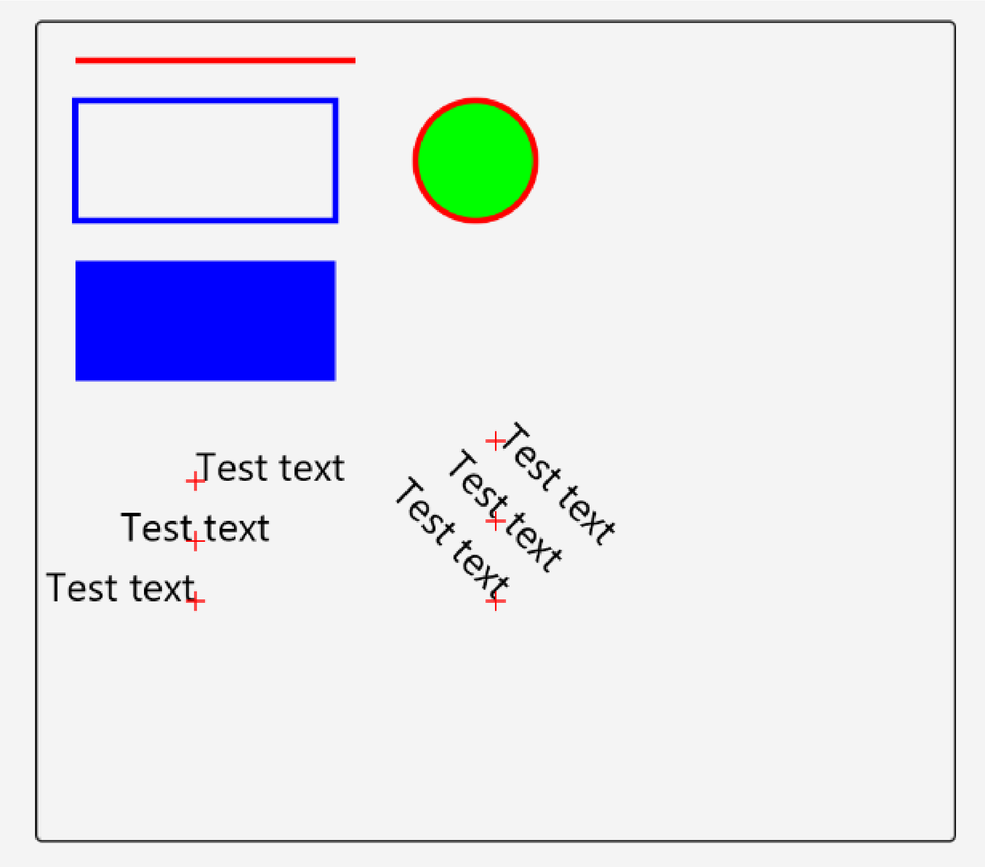
If we want to draw on different views at the same time, we need one Canvas for each view.

In the example programs we will use several drawing methods and draw onto the Activity and onto a Panel `pnlGraph` defined in the 'main' layout file. Here we need two canvases.

B4J
The Canvas is a node, it is not related to any other object like in B4A or B4i.
In the example program we'll use several drawing methods and draw onto the Canvas.

Only the XUI cross-platform project is explained!
It is much easier, even if it is for only one platform.

And the result.



7.1.1 Initialization B4XMainPage module

In the Class_Globals routine of the B4XMainPage module, we just initialize the Graphics custom view:

```
Sub Class_Globals
   Private Root As B4XView
   Private xui As XUI

   Private Graphics As GraphicsFirstSteps
End Sub
```

And call Graphics.Drawing in the B4XPage_Created routine.

```
Private Sub B4XPage_Created (Root1 As B4XView)
   Root = Root1
   Root.LoadLayout("Main")

   Graphics.Drawing
End Sub
```

7.1.2 Initialization GraphicsFirstSteps module

Then, in the Class_Globals routine of the GraphicsFirstSteps module, we declare a B4XView, the XUI library, the B4XCanvas and a B4XFont.
This example is made as a CustomView.

```
Sub Class_Globals
   Private mEventName As String 'ignore
   Private mCallBack As Object 'ignore

   Private xui As XUI
   Private xplGraph As B4XView
   Private cvsGraph As B4XCanvas
   Private xFont As B4XFont
End Sub
```

We initialize the B4XFont to draw the text, it is a cross platform xui method.

```
Public Sub Initialize (Callback As Object, EventName As String)
   mEventName = EventName
   mCallBack = Callback

   'define a default font with a size of 20
   xFont = xui.CreateDefaultFont(20)
End Sub
```

Here we set the Base object from the Designer to the B4XView xplGraph and ‘connect’ the B4XCanvas to the B4XView.

```
Public Sub DesignerCreateView (Base As Object, Lbl As Label, Props As Map)
   xplGraph = Base

   cvsGraph.Initialize(xplGraph)
End Sub
```

7.1.3 Draw a line

Then we draw a horizontal line onto the Activity or MainPage:
The method is:
DrawLine (x1 As Float, y1 As Float, x2 As Float, y2 As Float, Color as Int, StrokeWidth As Float)
Where:

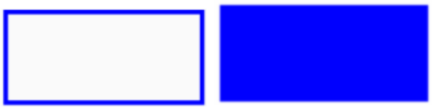
- x1, y1 are the coordinates of the start point in pixels
- x2, y2 are the coordinates of the end point in pixels
- Color is the line color
- StrokeWidth the line thickness in pixels

Then we draw a horizontal line onto pnlGraph with the same coordinates:
The coordinates are relative to the upper left corner of the view we draw on, the Panel `pnlGraph` in this case.

And the code:

```
' draw a horizontal line onto xplGraph
cvsGraph.DrawLine(20dip, 20dip, 160dip, 20dip, xui.Color_Red, 3dip)
```

7.1.4 Draw a rectangle

Then we draw an empty and a filled rectangle onto pnlGraph: 

The method is:
DrawRect (Rect1 As Rect, Color As Int, Filled As Boolean, StrokeWidth as Float)
Where:

- Rect1 is a rectangle object
- Color is the border or rectangle color
- Filled: False = only the border is drawn
  True = the rectangle is filled
- StrokeWidth is the line thickness of the border, not relevant when Filled = True

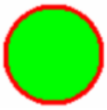
To draw a rectangle, we need a Rect object.
We:

- Define it with the name `rect1`.
- Initialize it with the coordinates of the upper left corner and the coordinates of the lower right corner.
- Draw it

And the code:

```
' draw a empty rectangle onto xplGraph
Private rect1 As B4XRect
rect1.Initialize(20dip, 40dip, 150dip, 100dip)
cvsGraph.DrawRect(rect1, xui.Color_Blue, False, 3dip)

' draw a filled rectangle onto xplGraph
rect1.Initialize(20dip, 120dip, 150dip, 180dip)
cvsGraph.DrawRect(rect1, xui.Color_Blue, True, 1dip)
```

7.1.5 Draw a circle

Then we draw filled circle with a border with a different color onto pnlGraph: 

The method is:

DrawCircle (x As Float, y As Float, Radius As Float, Color as Int, Filled As Boolean, StrokeWidth As Float)

Where:

- x, y are the coordinates of the center in pixels.
- Radius is the radius in pixels.
- Color is the border or circle color
- Filled: False = only the border is drawn True = the circle is filled
- StrokeWidth is the line thickness of the border, not relevant when Filled = True

There is no direct method to draw a filled circle with a border with a different color.
So, we first draw the filled circle and then the circle border in two steps.
Instead of using fixed values like 220dip we can also use variables like in the code below.
When a same value is used several times it's better to use variables because if you need to change the value you change it only once the value of the variable all the rest is changed automatically by the variable.

And the code:

```
' draw a filled circle with a border onto xplGraph
Private centerX, centerY, radius As Float
centerX = 220dip
centerY = 70dip
radius = 30dip
cvsGraph.DrawCircle(centerX, centerY, radius, xui.Color_Green, True, 3dip)
cvsGraph.DrawCircle(centerX, centerY, radius, xui.Color_Red, False, 3dip)
```

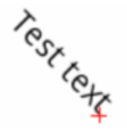
7.1.6 Draw a text

Then we draw a text. Test text

The method is:

DrawText (Text As String, x As Float, y As Float, Typeface1 As TypeFace, TestSize As Float, Color As Int Align1 As Align)
Where:

- Text is the text to draw
- x, y are the coordinates of the reference point (depending on the Align1 value) in pixels.
  The reference point is on the texts baseline.
- TypeFace1 is the text style
- TextSize is the text size in a typographic unit called 'point'.
  The text size is independent of the screen density!
  Don't use dip values!
- Color is the text color
- Align1 is the alignment of the text according to the refence point.
  Possible values: "LEFT", "CENTER", "RIGHT".

Then we draw a rotated text onto pnlGraph.  Test text

And we draw a cross on the reference point to show where it is and how the align does work.
The method is DrawTextRotated, it's the same as DrawText but with an additional parameter Degrees, the rotation angle.
Instead of using fiyed dip values in the routine we define three variables:

refX and refY	the coordinates of the reference point.
hl	the half of the cross-line length.

And the code:

```
   Private refX, refY, hl As Float
   refX = 80dip
   refY = 230dip
   hl = 5dip

   ' draw texts with three alignments onto xplGraph
   cvsGraph.DrawText("Test text", refX, refY, xFont, xui.Color_Black, "LEFT")
   DrawCross(refX, refY, hl)

'Draw a cross on the reference point
Private Sub DrawCross(x As Int, y As Int, l As Int)
   cvsGraph.DrawLine(x - l, y, x + l, y, xui.Color_Red, 1dip)
   cvsGraph.DrawLine(x, y - l, x, y +l, xui.Color_Red, 1dip)
End Sub
```

8 Simple drawing methods

In the second drawing program, SimpleDrawMethods, we use the other common drawing methods.

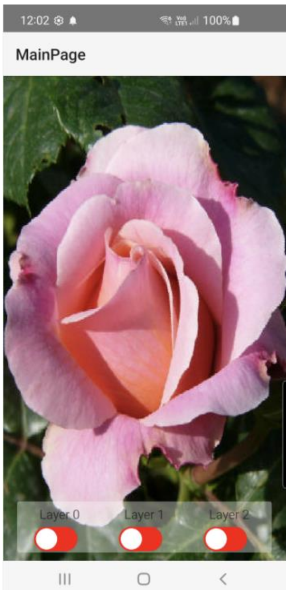
The project, a B4XPages XUI project using B4X objects like B4XView and B4XCanvas, is in the GraphicsSourceCode\SimpleDrawMethods directory.

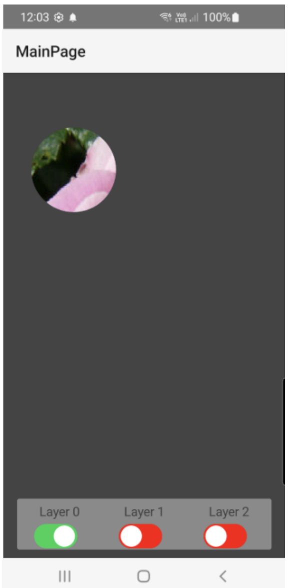
I left also older platform specific projects in the GraphicsSourceCode\SimpleDrawMethodsOld directory, one project for each platform: B4A, B4i, B4J.

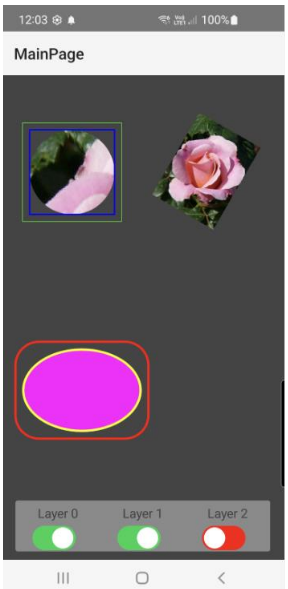
The program has no other purpose than to show what can be done with drawings.

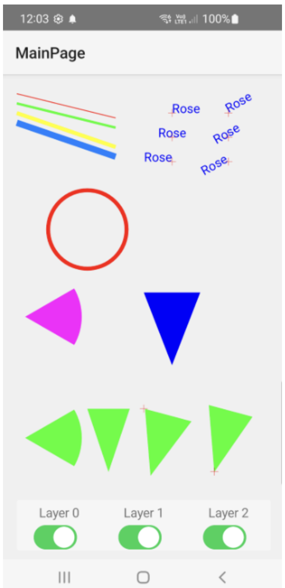
The program has four Panels. One for the background image and three layers which can be shown or hidden with B4XSwitches.

- pnlBackground: for the background image, a rose.
- pnlLayer(0) has a grey background and a movable transparent circle or square, letting see a part of the background image.
- pnlLayer(1) has a transparent color letting also see layer0. The transparent circle or square can still be moved.
- pnlLayer(2) has a light grey background hiding the layers behind it.









Background Layer 0 Layer 1 Layer 2

You can play with the switches to observe the different combinations of visible and hidden layers. And you can move the circle / square.

Only the B4A version is shown.

Analysis of the code:

There is one layout file for each project.

In the Class_Globals routine we declare the different objects and variables.

```
Sub Class_Globals
  Private Root As B4XView
  Private xui As XUI

  Private pnlBackground, pnlLayer0, pnlLayer1, pnlLayer2, pnlLayer(3) As B4XView
  Private cvsBackground, cvsLayer(3) As B4XCanvas
  Private xbmpBackground As B4XBitmap
  Private xRect, xRect0 As B4XRect
  Private xFont As B4XFont

  Private x100, y100, xc, yc, xc0, yc0, x1, y1, x2, y2, w1, w1_2, w2, w2_2, w2_21,
Radius As Double
End Sub
```

We have:

- 4 Panels / Panes as B4XView
- 4 Canvases as B4XCanvas
- 3 B4XSwitches from the XUI Views library.
- 2 B4XRect, rectangles used to draw rectangles.
- 1 B4XBitmap, holding the pnlBackground image.
- different variables used for the drawing.

Note that we use arrays of views for the three layer panels and canvases.

```
Private pnlLayer0, pnlLayer1, pnlLayer2, pnlLayer(3) As Panel.
Private cvsBackground, cvsLayer(3) As B4XCanvas
```

In the Sub B4XPage_Create routine we initialize the different views and add them to the activity:

```
Private Sub B4XPage_Created (Root1 As B4XView)
   Root = Root1
   Root.LoadLayout("Main")
   pnlLayer = Array As B4XView(pnlLayer0, pnlLayer1, pnlLayer2)

   'initialize the canvases
   cvsBackground.Initialize(pnlBackground) '
   For i = 0 To 2
       cvsLayer(i).Initialize(pnlLayer(i))
   Next

   'transfer the events to the underlying object
   'in B4J and B4i events are not transmitted automatically to the underlying object
   'when there is no event active routine for the object
#If B4J
   Private jo = pnlLayer(2) As JavaObject
   jo.RunMethod("setMouseTransparent", Array As Object(True))
   Private jo = pnlLayer(1) As JavaObject
   jo.RunMethod("setMouseTransparent", Array As Object(True))
#Else If B4i
  pnlLayer(2).As(Panel).UserInteractionEnabled = False
  pnlLayer(1).As(Panel).UserInteractionEnabled = False
#End If

   'initialize the background bitmap
   xbmpBackground = xui.LoadBitmap(File.DirAssets, "rose.jpg")

   Drawing
End Sub
```

We:

- Load the layout.
- Fill the pnlLayer array.
- Initialize the Canvases.
- Load the `rose.jpg` image file into the bitmap.
- Initialize the background image.
- Call the Drawing routine.

Sub Drawing we call the Drawing routine.

```
Private Sub Drawing
   'create the default font
   xFont = xui.CreateDefaultFont(16)

   'intialize the screen rectangle
   xRect0.Initialize(0, 0,pnlBackground.Width, pnlBackground.Height)

   'draw the background image
   cvsBackground.DrawBitmap(xbmpBackground, xRect0)
   cvsBackground.Invalidate

   'variables for 100%x and 100%y
   x100 = pnlLayer(0).Width
   y100 = pnlLayer(0).Height

   DrawLayer0
   DrawLayer1
   DrawLayer2
End Sub
```

The DrawLayer routines are hopefully enough self-explanatory.

The stwLayer_ValueChanged routine. Nothing special.

```
' B4XSwitch VakueChanged event routine, all three B4XSwitches call this routine
Private Sub stwLayer_ValueChanged (Value As Boolean)
   Dim stw As B4XSwitch
   Dim index As Int

   stw = Sender
   index = stw.Tag
   pnlLayer(index).Visible = Value
End Sub
```

The pnlLayer0_Touch routine.

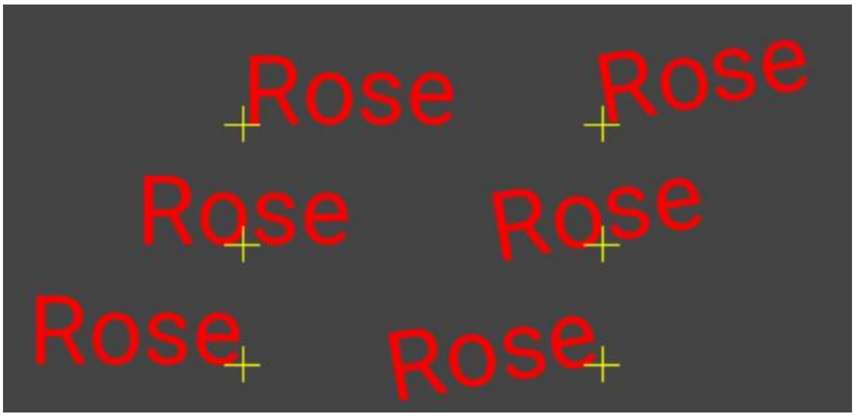
```
Private Sub pnlLayer0_Touch (Action As Int, X As Float, Y As Float)
  Select Action
    Case pnlLayer0.TOUCH_ACTION_MOVE
      'redraws the background of the transparent part
      xRect.Initialize(xc0 - w2_21, yc0 - w2_21, xc0 + w2_21, yc0 + w2_21)
      cvsLayer(0).DrawRect(xRect, xui.Color_DarkGray, True, 1)

      xc0 = X
      yc0 = Y
      xRect0.Initialize(xc0 - w2_2 , yc0 - w2_2, xc0 + w2_2, yc0 + w2_2)
      'draws the new transparent part
    #If B4J
      cvsLayer(0).ClearRect(xRect)
    #Else
      cvsLayer(0).DrawCircle(xc0, yc0, w2_2, xui.Color_Transparent, True, 1dip)
    #End If
  cvsLayer(0).Invalidate
  End Select
End Sub
```

In B4A and B4i when we draw with a transparent color, the objects background becomes transparent. Drawing with a transparent color does nothing in B4J. We can set the background to transparent only with the ClearRect method, therefor a square instead of a circle.

Looking closer on the displayed texts we see the reference point for each text.

```
  cvsLayer(2).DrawText("Rose", x1, y1, Typeface.DEFAULT,16,Colors.Red,"LEFT")
  DrawCross(x1, y1, Colors.Yellow)
```

These are the x1 and y1 coordinates used to display the texts.



LEFT alignment.

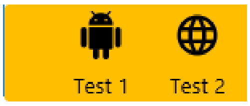
CENTER alignment.

RIGHT alignment.

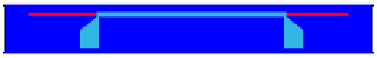
9 Other examples in the B4X CustomViews booklet

There are other XUI examples in the B4X CustomViews booklet.

Chapter 6 XUI xCustomButton



Chapter 7 XUI xLimitBar



10 Other examples from the forum

Several examples are published in the forum.

To search on the forum use B4X XUI as a prefix to filter the results.
All the projects below have a same *.bas file for all three products.

A few examples:

CircularProgressbar	custom view class.
xChart	custom view class and B4XLibrary.
AnotherDatePicker	included as B4XDateTemplate in the XUI Views B4XLibrary.
xRotaryKnob	custom view class and B4XLibrary.
xGauges	custom view class and B4XLibrary.
xGraph	custom view class and B4XLibrary.
xResizeAndCrop	custom view class and B4XLibrary.

11 Libraries

It is possible to compile modules into libraries.

As we have seen, custom views for example, have the same module *bas file.
But, if you want to compile those into product specific libraries, you need to compile one library for each product!
This is needed because the library code is different for each operating system.

A better method is to compile it into a b4xlib library.
More details in the B4X libraries chapter in the B4X Language booklet.

11.1 B4X libraries *.b4xlib

B4X libraries are cross platform libraries introduced in B4A 8.80, B4i 5.50 and B4J 7.00.

These libraries contain cross platform classes which don’t need to be compiled as libraries.

A B4X library is a simple zip file with the following structure:

- Code modules. All types are supported including Activities ans Services.
- Files, including layout files.
- Optional manifest file with the following fields:
  - Version
  - Author
  - DependsOn (list of required libraries), Supported Platforms. Fields can be shared between the platforms or be platform specific.

Files and code modules can also be platform specific.

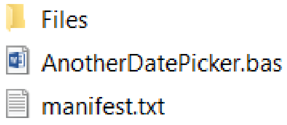
Creating a b4x library is very simple. You just need to create a zip file with these resources. The zip file extension should be b4xlib. That's all.

Note that the source code can be extracted from a b4x library.

b4x libraries appear like all other libraries in the Libraries tab.

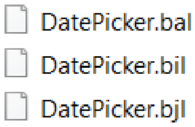
Example: The AnotherDatePicker.b4xlib

The zip file structure:



Files
AnotherDatePicker.bas
manifest.txt

Files contains all the needed files, the three layout files in the example.



DatePicker.bal
DatePicker.bil
DatePicker.bjl

AnotherDatePicker.bas is the crossplatform Custom View file.
Manifest.txt contains:

`Version=2.00`	version number.
`Author=Erel`	version number.
`B4J.DependsOn=jXUI, jDateUtils`	libraries used for B4J.
`B4A.DependsOn=XUI, DateUtils`	libraries used for B4A.
`B4i.DependsOn=iXUI, iDateUtils`	libraries used for B4i.

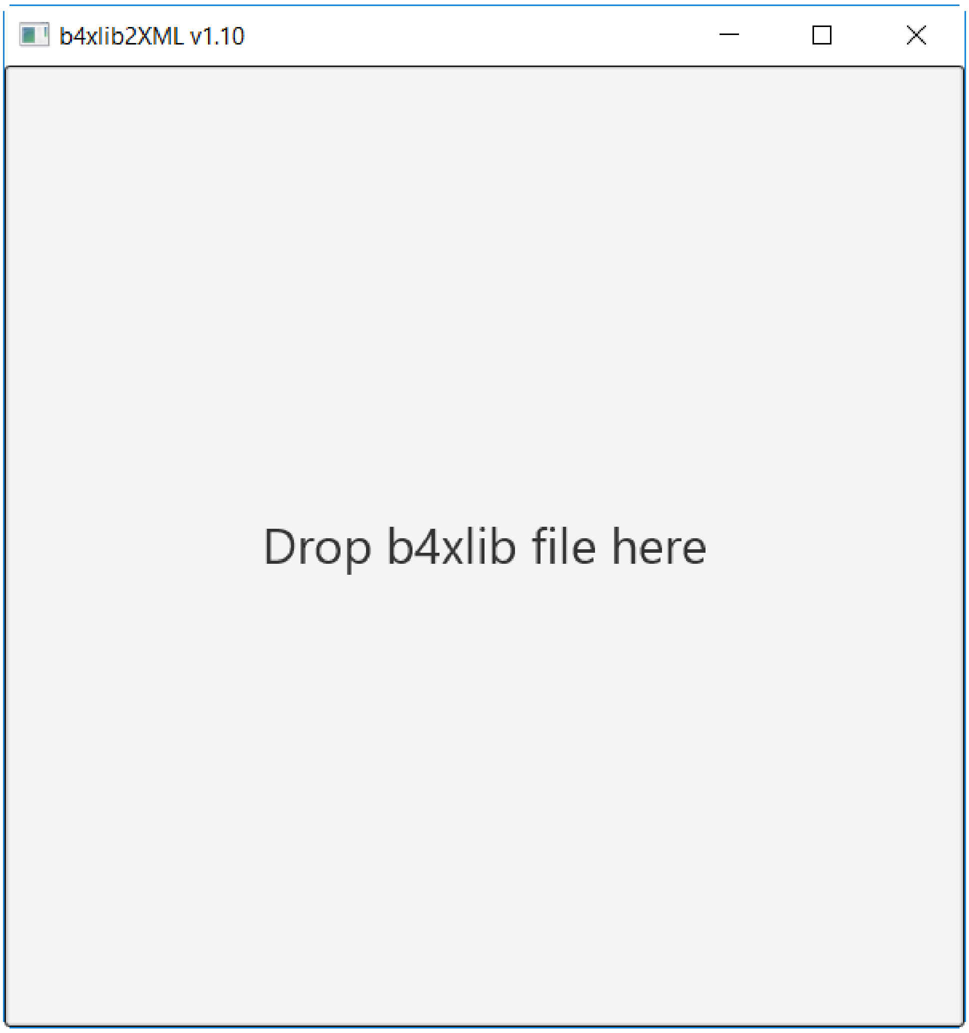
Copy the xxx.b4xlib file to the AdditionalLibaries\B4X folder.

11.1.1 Xml help files for B4X Libraries

Erel has written an application to create xml help files for B4X Libraries.

You can download it from HERE.

It looks like this:



Simply, drag and drop a xxx.b4xlib file into the from.
The xml file will be created, and you will be asked where you want to save it.

Tip:
Save all the b4xlib xml files into a specific folder.
Example: AdditionalLibaries\B4XLibraryXMLFiles.

The xml files are useful for the HelpViewer applications like:
B4X Help Viewer
B4X Object Browser