### Che cos’è un’immagine Raster? Per immagini raster (o bitmap) si intendono le immagini fotografiche (scattate per esempio con lo smartphone) che possono essere in formato JPG, PNG, GIF oppure TIFF. Queste tipologie di immagini sono composte da pixel colorati, ovvero singoli puntini invisibili che compongono una figura. Le immagini raster hanno una dimensione e risoluzione propria quindi, nel momento in cui vengono ingrandite oltre le loro dimensioni originali, subiscono perdite di dettagli e appaiono sgranate o dentellate. Aumentando infatti le dimensioni di un’immagine raster si manifesta un effetto visivo molto sgradevole chiamato pixellatura. Il principale vantaggio di un’immagine raster però è che grazie ai milioni di pixel che la compongono, può rappresentare svariati dettagli. Pensa a quando scatti una fotografia, grazie a milioni di pixel puoi catturare qualsiasi dettaglio. Che cosa’è un’immagine vettoriale? Le immagini vettoriali invece sono descritte mediante un insieme di primitive geometriche che definiscono punti, linee, curve e poligoni. il formato vettoriale è definito attraverso equazioni matematiche ed è indipendente dalla risoluzione, infatti può essere ingrandito all’infinito senza subire perdite di qualità e definizione. Per esempio, un logo in vettoriale (composto da tracciati) può essere utilizzato sia su un biglietto da visita, sia su un cartellone pubblicitario mantenendo i dettagli invariati. Il formato vettoriale presenta inoltre un’alta efficienza nella dimensione del file, infatti il peso di un documento è inferiore rispetto ad uno raster e può essere inviato da un computer all’altro più rapidamente.

### Supporto Android Studio per vector drawables

In Android studio 1.4 viene introdotto un tool per prendere un vector asset from material design icon library, o convertire file svg in vector drawables

Su qualsiasi folder nell’applicazione, faccio click destro, New, Vector Asset, da qui vedo la finestra di vector asset studio. Da qui posso prendere un vector asset da material design icon library

1. Voglio scegliere dalla libreria di material icon: faccio click su Choose button. Mi si apre una finestra con tutti gli assets della libreria, ne scelgo una e clicco ok. Co sono delle opzioni in alto a destra per cambiare altezza, larghezza, opacità, automirroring (versione 21 in poi) ecc…
2. Se voglio utilizzare un mio file svg, clicco su sfoglia e scelgo il mio file

In entrambi i casi vedo un’anteprima dell’immagine scelta.

In questo strumento non sono ancora supportate alcune funzionalità, nel caso in cui venga trovato qualcosa non supportato nel file svg, verrà riportato come errore in basso nella finestra, con annessi error details.

Quando ho finito e sono soddisfatto dell’anteprima 🡪 Next.

Nella pagina successiva posso vedere il file SML nella mia directory web/drawable. 🡪 Finish.

Sotto la directory “generated” c’è una directory “pngs” in cui avrò cartelle con diversi DPI e numeri di versione. Vengono create due directory per un’immagine, la prima contiene il file .png, la seconda di conseguenza contiene il .xml.

COMPATIBILITÀ

Se l’app ha come target versioni Lollipop o più recenti, tutta l’API spec è supportata.

Se la versione di sdk è < 21 ho alcune limitazioni:

* I tag <group> e <clip-path> non sono supportati
* Posso avere solo file png statici: nel file SML del vector drawable non supporto attributi dinamici (theming, auto mirroring..)
* Non supportate le resources references ad altri file SML.

Inoltre,la finestra di anteprima in Android Studio usa un meccanismo di display diverso da quello in vector asset studio.

Infine, se devo typecastare gli oggetti drawable nel codice java, devo castarli come bitmap drawable invece di vector drawable (solo in dispositivi pre Lollipop).

++++++++++++++++++++++++++++++++++++++++++++++++

Android Studio include uno strumento chiamato Vector Asset Studio che serve ad aggiungere icone materiali e importare file SVG (Scalable Vector Graphic) e PSD (Photoshop Document) nel proprio progetto come risorse vector drawables. Usare questi invece che i bitmaps serve a ridurre la dimensione del proprio APK perché lo stesso file può essere ridimensionato per diverse densità dello schermo, senza rinunciare alla qualità dell’immagine.

### Riguardo Vector Asset Studio

Vector Asset Studio aggiunge una grafica vettoriale al progetto con un file XML che descrive l’immagine.

Le versioni da Android 4.4 (API level 20) e più vecchie non supportano i vector drawables. In questi casi con Vector Asset Studio si può solo generare file PNG (Portable Network Graphic) o usare la libreria di supporto.

Per la retrocompatibilità, Vector Asset Studio genera immagini bitmap del vector drawable. I drawable sono nello stesso package nell’APK.

Per fare riferimento ai vector drawables uso Drawable nel codice Java oppure @drawable in XML.

### Tipi di grafica vettoriale supportati

Google Material Design offre icone materiali che si possono usare nelle applicazioni Android. Vector Asset Studio aiuta l’utente a scegliere, importare e dimensionare queste icone, per poi poter definire l’opacità e anche le impostazioni di mirroring.

Con Vector Asset Studio, inoltre, l’utente può importare i propri file SVG e PSD, anche se non ne supporta tutti gli aspetti. Una volta scelto il file, Vector Asset Studio fornisce un feedback immediato se il codice grafico è supportato o no. Poi converte il file in un XML che contiene codice VectorDrawable. Se si ricevono errori bisogna verificare che il vector drawable sia raffigurato nel modo previsto.

Per versioni Android 5.0 (API level 21) e più recenti, si può usare la classe AnimatedVectorDrawable per, appunto, animare le proprietà della classe VectorDrawable.

### Considerazioni su file SVG e PSD

Un file SVG è un file che usa un formato grafico vettoriale bidimensionale. Descrive l’immagine usando un formato di testo, basato su XML. I file SVG sono sviluppati come formato standard per mostrare grafiche vettoriali sul web.

Il PSD è un formato di file proprietario sviluppato da Adobe Systems. La sua particolarità è quella di essere un formato di grafica vettoriale molto versatile. È utilizzato per creare immagini vetoriali e progetti multimediali.

Il vector drawable è appropriato per icone semplici, mentre quelle con troppi dettagli funzionano meglio come immagini bitmap.

Il caricamento iniziale di un vector drawable può costare di più in termini di cicli di CPU rispetto alla corrispondente immagine bitmap. Mentre successivamente l’uso di memoria è simile tra le due.

Se l’immagine da vettorizzare è troppo grande, caricarla potrebbe richiedere troppo tempo (best practice: limitare le dimensioni).

Sebbene i vector drawables supportino anche più di un colore, spesso è raccomandabile usare icone nere (android:fillColor=”#FF000000”). In questo modo è possibile aggiungere una tinta, che diventerà il colore dell’icona. Se il colore dell’icona non è nero, il colore potrebbe fondersi con quello della tinta.

### Generazione di PNG

Per le versioni precedenti ad Android 5.0, Vector Asset Studio aggiunge il file vector drawable al progetto, inoltre Gradle crea immagini PNG bitmap a varie risoluzioni. Le densità delle immagini PNG sono specificate dalla proprietà generatedDensities del Domain Specific Language in un file build.gradle.

Per le versioni successive ad Android 5.0, Vector Asset Studio supporta tutti gli elementi VectorDrawable.

Per la retrocompatibilità con Android 4.4 e precedenti, Vector Asset Studio supporta i seguenti elementi XML:

<vector>

* android:width
* android:height
* android:viewportWidth
* android:viewportHeight
* android:alpha

<group>

* android:rotation
* android:pivotX
* android:pivotY
* android:scaleX
* android:scaleY
* android:translateX
* android:translateY

<path>

* android:pathData
* android:fillColor
* android:strokeColor
* android:strokeWidth
* android:strokeAlpha
* android:fillAlpha
* android:strokeLineCap
* android:strokeLineJoin
* android:strokeMiterLimit

### Libreria di Supporto

Prima di usare Vector Asset Studio, bisogna aggiungere una dichiarazione nel file build.gradle:

android {  
  defaultConfig {  
    vectorDrawables.useSupportLibrary = true  
  }  
}  
  
dependencies {  
  compile 'com.android.support:appcompat-v7:23.2.0'  
}

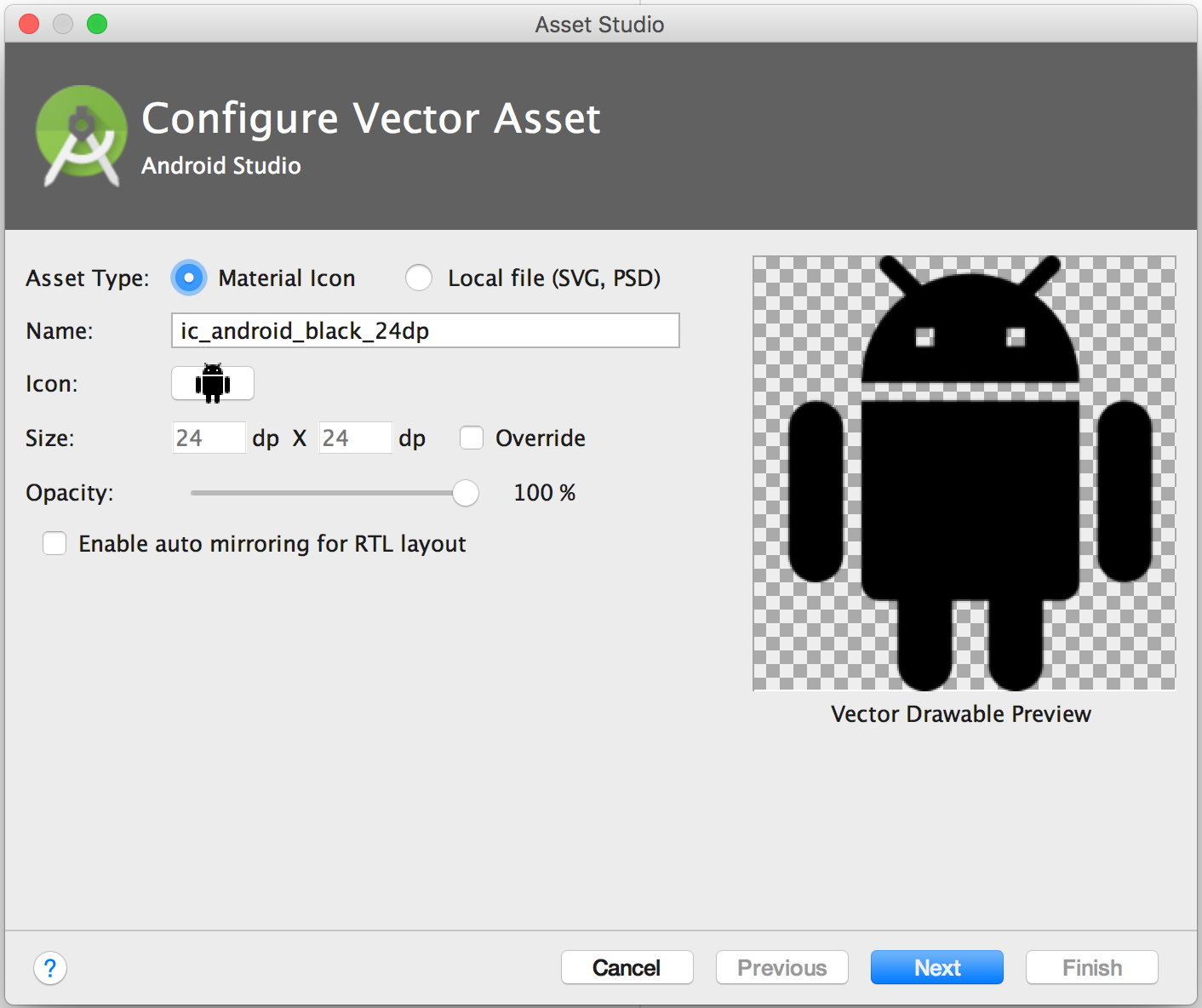
Inoltre bisogna usare tecniche di codifica compatibili con la Libreria di Supporto, come per esempio l’uso dell’attributo app:srcCompat invece che l’attributo android:src per i vector drawables.

-----------------Running Vector Asset Studio--------------------------------------------------------------------

Per avviare Vector Asset Studio:

* 1. In Android Studio, aprire un Android app project
  2. In *Project window*, selezionare Android view
  3. Click destro sul folder res e selezionare **New** > **Vector** **Asset**

Apparirà la schermata di Vector Asset Studio:



**Sdisaj:**

* 1. **Se appare la finestra *Need newer Android plugin for Gradle*, correggere la versione del Gradle:**
     1. **Seleziona File > Project Structure**
     2. **Seleziona Project**
     3. **Nel campo Android Plugin Version cambia la versione del plugin Android per Gradle alla 1.5.0 o più recente, poi clicca OK**
     4. **Nella Android view della *Project window*, clicca con il tasto destro il folder res e seleziona New > Vector Asset**
  2. Importa la grafica vettoriale

## Importare una grafica vettoriale

## VAS aiuta a importare grafiche vettoriali nel progetto seguendo una delle seguenti procedure:

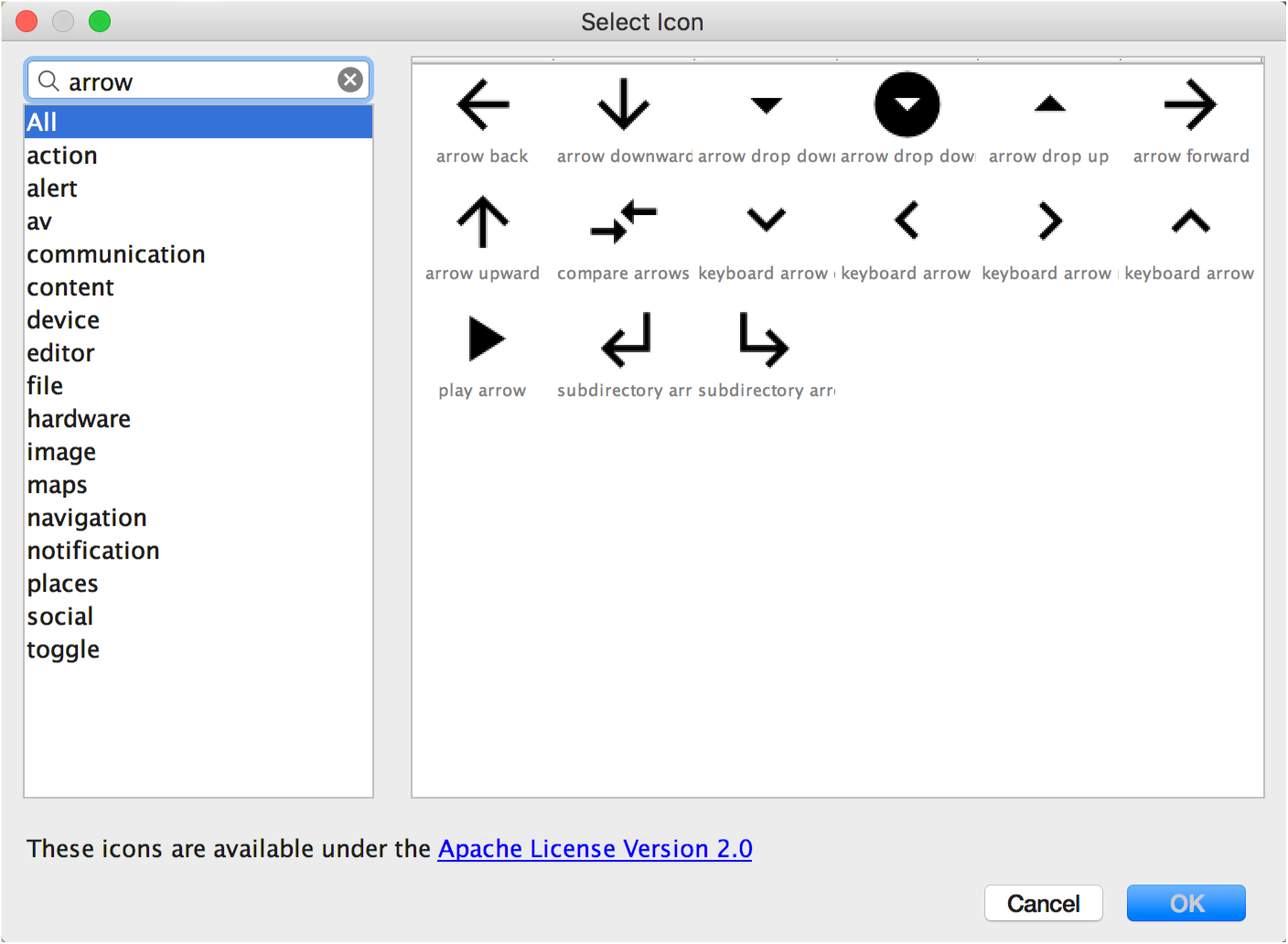
## Aggiungere una material icon

## Importare file SVG o PSD

### Aggiungere material icon

Dopo che hai aperto Vector Asset Studio, puoi aggiungere una material icon come segue:

1. In VAS seleziona Material Icon
2. Nel campo icona, clicca il bottone
3. Apparirà la finestra per selezionare l’icona. Puoi selezionare le icone in base alle categorie che appaiono nella lista a sinistra.



Seleziona una material icon e clicca OK. L’icona apparirà nella Vector Drawable Preview.

1. È possibile cambiare nome, grandezza, opacità e mirroring right to left:

**nome**: scrivi un nuovo nome se non vuoi usare quello predefinito. VAS crea automaticamente un nome unico (eventualmente aggiungendo un numero alla fine del nome).

**Override**: selezionare se si vuole cambiare la grandezza dell’immagine. La grandezza predefinita è 24x24 dp.

**Opacità**: usa il cursore per cambiare l’opacità dell’immagine.

**Abilita auto mirroring RTL**: selezionare se si vuole un’immagine specchiata da mostrare quando il layout è da destra a sinistra (RTL), invece che da sinistra a destra.

1. Clicca su Next
2. Puoi cambiare il modulo e la directory delle risorse:

**Directory res**: seleziona il set di risorse in cui si vuole aggiungere il vector drawable: src/main/res, src/debug/res, src/release/res, oppure un set scelto dall’utente.

Per definire un nuovo set, selezionare File > Project Structure > app > Build Types.

1. Clicca Finish

VAS aggiungerà un file XML al progetto, il quale definisce il vector drawable. Questo file viene messo nel folder app/src/main/res/drawable/.

Dall’Android view della Project window, è possibile vedere il file XML generato nel folder drawable.

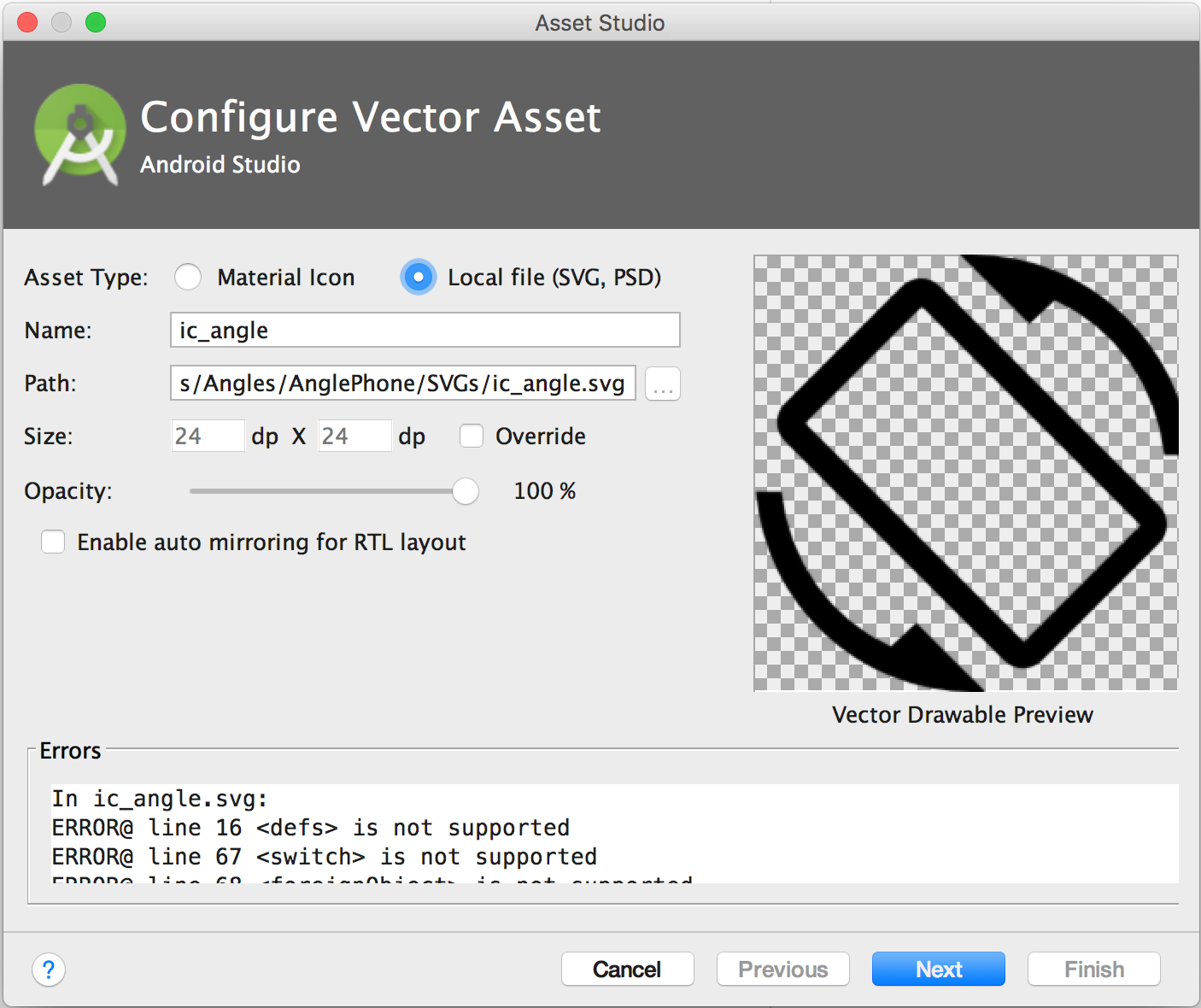
1. Costruisci il progetto

Se il livello minimo dell’API è Android 4.4 (API level 20) o più basso, e non è abilitata la libreria di supporto, VAS genera file PNG. Dalla Project Files view della Project window è possibile vedere il file PNG generato e i file XML nel folder app/build/generated/res/pngs/debug/ .

Best Practice: non modificare questi file generati, piuttosto lavorare sul file XML. (?????)

### Importare file SVG o PSD

1. Selezionare Local file. (Il file deve essere salvato sul dispositivo corrente)
2. Cliccare l’immagine desiderata. L’immagine apparirà nell’anteprima Vector Drawable.



Se il file selezionato contiene delle caratteristiche non supportate, un errore apparirà alla base di Asset Studio, come mostrato in figura.

Se ci sono errori, assicurati che il vettore grafico selezionato “renderizzi” appropriatamente.

Scorri la lista per visualizzare gli errori.

1. È possibile cambiare nome, grandezza, opacità e mirroring right to left:

**nome**: scrivi un nuovo nome se non vuoi usare quello predefinito. VAS crea automaticamente un nome unico (eventualmente aggiungendo un numero alla fine del nome).

**Override**: selezionare se si vuole cambiare la grandezza dell’immagine. La grandezza predefinita è 24x24 dp.

**Opacità**: usa il cursore per cambiare l’opacità dell’immagine.

**Abilita auto mirroring RTL**: selezionare se si vuole un’immagine specchiata da mostrare quando il layout è da destra a sinistra (RTL), invece che da sinistra a destra.

1. Clicca su Next
2. Puoi cambiare il modulo e la directory delle risorse:

**Directory res**: seleziona il set di risorse in cui si vuole aggiungere il vector drawable: src/main/res, src/debug/res, src/release/res, oppure un set scelto dall’utente.

Per definire un nuovo set, selezionare File > Project Structure > app > Build Types.

1. Clicca Finish

VAS aggiungerà un file XML al progetto, il quale definisce il vector drawable. Questo file viene messo nel folder app/src/main/res/drawable/.

Dall’Android view della Project window, è possibile vedere il file XML generato nel folder drawable.

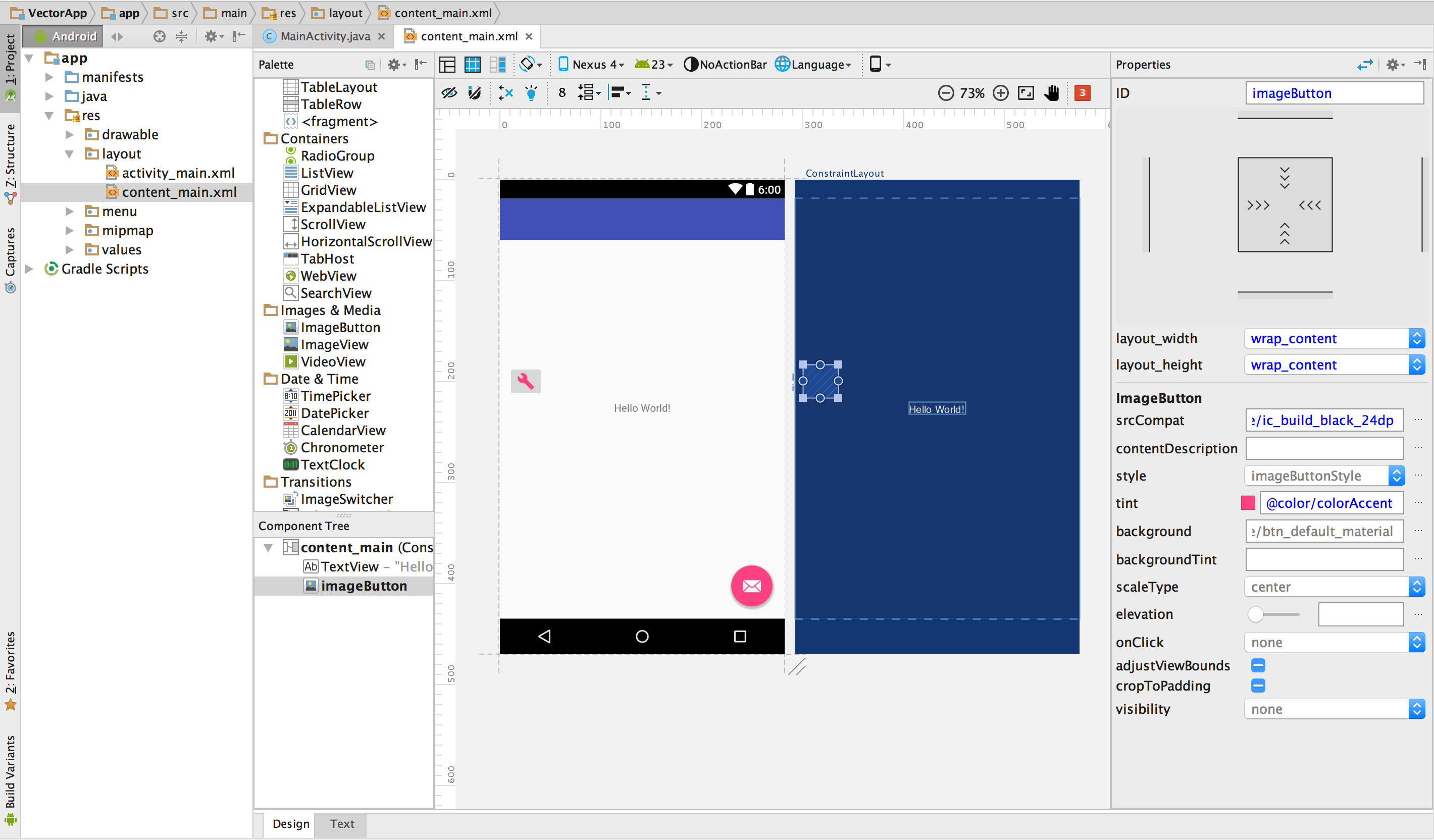
1. Costruisci il progetto

Se il livello minimo dell’API è Android 4.4 (API level 20) o più basso, e non è abilitata la libreria di supporto, VAS genera file PNG. Dalla Project Files view della Project window è possibile vedere il file PNG generato e i file XML nel folder app/build/generated/res/pngs/debug/ .

Best Practice: non modificare questi file generati, piuttosto lavorare sul file XML. (?????)

## Aggiungere un vector drawable al layout

## Nel layout è possibile impostare qualsiasi widget relativo alle icone (come ImageButton, ImageView ecc…), per indicare un vector drawable. Il seguente layout mostra un vector drawable su un bottone:



Per mostrare un vector drawable su un widget, come in figura:

1. Aprire il progetto e importare un vector drawable
2. Nella Android view della Project window, fare doppio click sul file layout XML
3. Clicca l’icona Design per mostrare l’editor di layout
4. Trascina un ImageButton dalla Palette nell’editor di layout
5. Nella finestra Resources, seleziona Drawable nel pannello di sinistra, poi seleziona il vector drawable importato e clicca OK

Il vector drawable apparirà nell’ImageButton nel layout

1. Per cambiare il colore dell’immagine nell’accent color, vai su Proprietà e clicca sui tre puntini accanto a Tint
2. Nella finestra Resources, clicca su Color nel pannello sinistro e seleziona colorAccent, poi clicca OK

Se il progetto usa una libreria di supporto, il codice dovrebbe essere così:

<ImageButton  
  android:layout\_width="wrap\_content"  
  android:layout\_height="wrap\_content"  
  app:srcCompat="@drawable/ic\_build\_black\_24dp"  
  tools:layout\_editor\_absoluteX="11dp"  
  tools:layout\_editor\_absoluteY="225dp"  
  android:id="@+id/imageButton"  
  android:tint="@color/colorAccent" />

Se, invece, il progetto non usa alcuna libreria di supporto, il codice del vector drawable sarà di questo tipo:

android:src="@drawable/ic\_build\_black\_24dp"

## Fare riferimento a un vector drawable nel codice

## Nella maggior parte dei casi, ci si può riferire ai vector drawables con “@drawable” (per XML) o “Drawable” (in codice Java).

## Per esempio, lo snippet seguente di codice XML applica l’immagine a una view:

<ImageView  
    android:layout\_height="wrap\_content"  
    android:layout\_width="wrap\_content"  
    android:src="@drawable/myimage" />

* E il seguente codice Java richiama l’immagine come Drawable:

Resources res = getResources();

Drawable drawable = res.getDrawable(R.drawable.myimage, getTheme());

Il metodo getResources() risiede nella classe Context, che è applicata agli oggetti di UI, come attività, frammenti, layout, view ecc…

* Se l’app usa la libreria di supporto, ci si può riferire a un vector drawable con uno statement app:srcCompat:

<ImageView  
    android:layout\_height="wrap\_content"  
    android:layout\_width="wrap\_content"  
    app:srcCompat="@drawable/myimage" />

* Occasionalmente, si può avere il bisogno di typecastare la risorsa drawable alla sua classe esatta, come quando serve usare caratteristiche specifiche della classe VectorDrawable. Per farlo, è possibile usare il codice Java seguente:

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.LOLLIPOP) {  
   VectorDrawable vectorDrawable = (VectorDrawable) drawable;  
} else {  
   BitmapDrawable bitmapDrawable = (BitmapDrawable) drawable;  
}

Si può accedere alle risorse vector drawable solo dal thread principale.

## Modificare codice XML generato da Vector Asset Studio

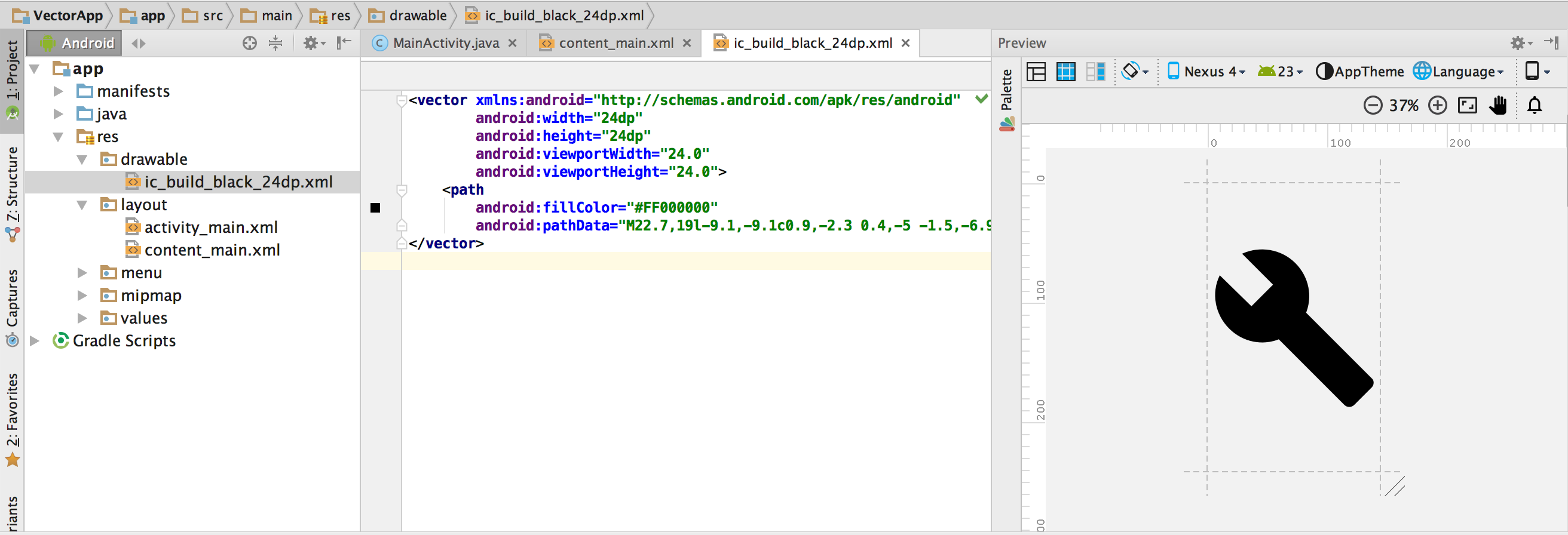
## È possibile modificare il codice XML del vector drawable, ma non i file PNG e i corrispondenti codici XML generati a tempo di costruzione. In ogni caso, non è una pratica raccomandata.

## Quando si usa la tecnica di generazione PNG, VAS si assicura che il vector drawable corrisponda al PNG, e che il manifest contenga il codice corretto. Se si aggiunge codice che non è supportato su Android 4.4 (API level 20) o precedenti, il vettore e le immagini PNG potrebbero essere differenti.

## Inoltre bisogna assicurarsi che il manifest contenga codice che supporti i cambiamenti.

## Per modificare il file XML quando non si usa la tecnica della libreria di supporto:

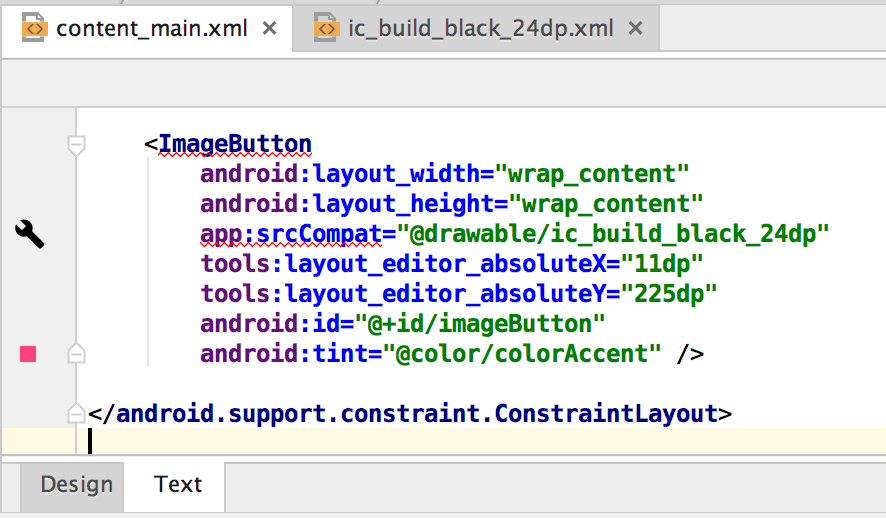
## Nella project window, fare doppio click sul file XML generato nel folder drawable.



1. Modificare il codice XML basandosi su cosa è supportato dal livello minimo di API:
   * Per Android 5.0 (API level 21) e superiori, VAS supporta tutti gli elementi Drawable e VectorDrawable. È possibile aggiungere elementi XML e cambiare i valori.
   * Per Android 4.4 (API level 20) e inferiori, VAS supporta tutti gli elementi drawable e un sottoinsieme di elementi VectorDrawable. È possibile cambiare i valori nel codice generato e aggiungere elementi XML che sono supportati.
2. Buildare il progetto e controllare che il vector drawable e le immagini raster corrispondenti appaiano allo stesso modo.

Ricordare che i file PNG generati potrebbero apparire diversamente da come mostrati nell’anteprima a causa di diversi motori di rendering e per qualche cambiamento sul vector drawable prima del Build.

Se si aggiunge codice al file XML creato da VAS, qualsiasi aspetto non supportato da Android 4.4 e inferiori non apparirà nel file PNG generato. Di conseguenza, quando si aggiunge codice, bisogna sempre controllare che il PNG generato corrisponda al vector drawable. Per farlo, si può fare doppio click sull’immagine PNG mostrata nel margine sinistro dell’editor di codice, quando il codice si riferisce al drawable.



## Rimuovere un vector drawable da un progetto

## Per rimuovere un vector drawable da un progetto:

## Nella project window, eliminare il file XML generato selezionando il file e premendo il tasto Canc (o selezionare Modifica > Cancella). Si aprirà la finestra di eliminazione sicura

## Facoltativamente selezionare le opzioni per trovare dove viene utilizzato il file nel progetto e fare click su OK.

## Android Studio elimina il file dal progetto e dal drive. Però, se si è scelto di cercare dove il file è usato nel progetto, e vengono trovati utilizzi, è possibile controllarli e decidere se cancellare il file

## Seleziona Build > Clean Project

## Qualsiasi file PNG e XML autogenerato, corrispondente al vector drawable cancellato, verrà rimosso dal progetto e dal drive

Per la pratica, vedere anche questo sito:

https://www.homeandlearn.co.uk/android/android\_vector\_assets.html