# AppCross Interactive Object Library Specification

Version 0.0.1

內容

[AppCross Interactive Object Library Specification 1](#_Toc385338211)

[Version 0.0.1 1](#_Toc385338212)

[1. 概述 3](#_Toc385338213)

[2. Interactive Object Class 3](#_Toc385338214)

[2.1 Catalog 3](#_Toc385338215)

[2.2 Build Interactive Object Library 8](#_Toc385338216)

[3. Interactive Object Library Description 11](#_Toc385338217)

[3.1 Bitmap Handler 11](#_Toc385338218)

[3.2 Clear Cache 12](#_Toc385338219)

[3.3 XML Parser 12](#_Toc385338220)

[3.4 Device 13](#_Toc385338221)

[3.5 Event Handler 13](#_Toc385338222)

[3.6 Event Message 13](#_Toc385338223)

[3.7 File Handler 15](#_Toc385338224)

[3.8 Sqlite Handler 16](#_Toc385338225)

[3.9 Thread Handler 17](#_Toc385338226)

[3.10 Data Type 17](#_Toc385338227)

[3.11 Version Handler 18](#_Toc385338228)

[3.12 Word Handler 18](#_Toc385338229)

[3.13 Zip Handler 18](#_Toc385338230)

## 概述

AppCross Interactive Object Library 主要提供Android AppCross Reader程式互動元件，在設計上每個互動元件設計成獨立的View延伸物件，有些物件會有群組關係，有群組關係的物件會透過群組控制模組來管理，每個互動元件都會依據JSON描述檔來顯示位置跟大小還有屬性。

目前所有的互動元件皆包含在InteractiveView Android Project裡，在此Project內亦提供互動元件會用到的Animation模組、JSON Parser模組、Handler模組、Global模組。

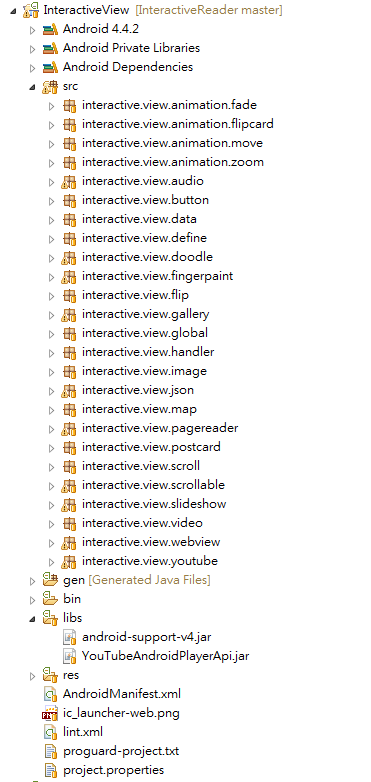
## Interactive Object Class

### Catalog

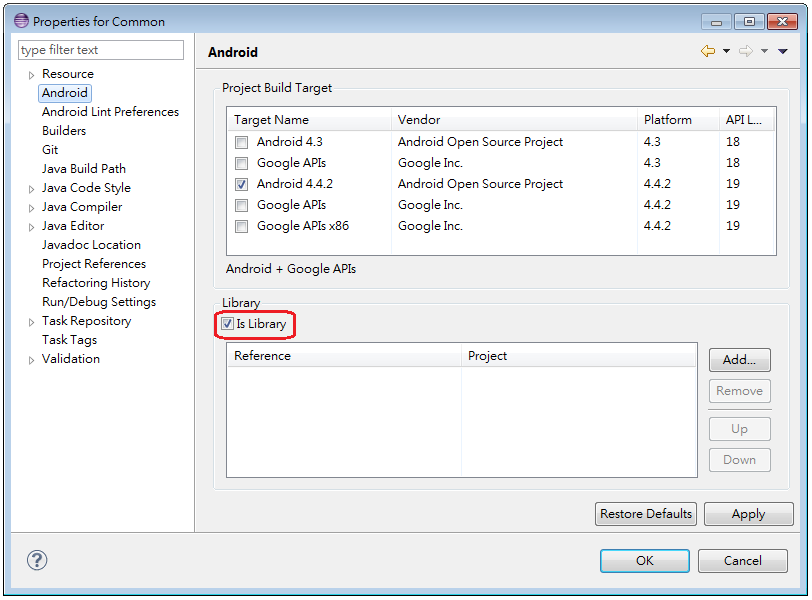
|  |  |  |
| --- | --- | --- |
| **Module Name** | **File** | **Description** |
| Animation Fade | FadeHandler.java | 控制兩個View切換時，淡入淡出的效果 |
| Animation Rotate | Rotate3dAnimation.java  Rotate3d.java | 控制視窗翻轉的動畫效果 |
| Animation Move | MoveHandler.java | 控制視窗移動的動畫效果 |
| Animation Zoom | ZoomHandler.java | 控制視窗zoom in和zoom out動畫效果 |
| Audio Player | AudioPlayer.java  MediaController.java | 聲音播放與播放控制列模組 |
| Interactive Button | ButtonView.java | Button互動元件 |
| Page Data | PageData.java | 頁面資訊模組 |
| Define | InteractiveDefine.java | 互動元件屬性定義檔 |
| Interactive Doodle | DoodleView.java | 繪圖視窗互動元件 |
| Finger Paint | FingerPaintView.java | 繪圖功能模組 |
| Flipper Show | FlipperView.java  AnimationType.java | 控制視窗滑動顯示 |
| Gallery View | GalleryView.java | Gallery視窗 |
| Global | Global.java | 全域共用函數 |
| Data type | Type.java | Define data type |
| Group Object Handler | InteractiveAudioHandler.java  InteractiveButtonData.java  InteractiveButtonHandler.java  InteractiveEventData.java  InteractiveGoogleMapData.java  InteractiveHandler.java  InteractiveImageData.java  InteractiveMediaData.java  InteractiveMediaHandler.java  InteractiveMediaLayout.java | 群組物件控制模組，控制Button元件與Image元件的行為，控管Video單一播放的行為 |
| Interactive Image | EventImageView.java  ImageViewData.java  ImageViewHandler.java | 事件圖片與手勢型為控制 |
| JSON Object | InteractiveAudio.java  InteractiveButton.java  InteractiveDoodle.java  InteractiveIframe.java  InteractiveImage.java  InteractiveMap.java  InteractiveObject.java  InteractivePostcard.java  InteractivePuzzle.java  InteractiveScrollable.java  InteractiveSlideshow.java  InteractiveVideo.java  InteractiveWebPage.java | 個互動元件的Json parser |
| Interactive Google map | GoogleMapActivity.java  GoogleMapFragment.java  GoogleMapView.java | Google map互動元件 |
| Reader Page Handler | ChapterChangeListener.java  ChaptersAdapter.java  DisplayPage.java  HorizonPageView.java  PageChangeListener.java  PageContainer.java  PagerAdapter.java  PageReader.java  PagesAdapter.java  VerticalPagerAdapter.java  VerticalPageView.java  VerticalViewPager.java  ViewPager.java | Reader的頁面控制模組，包含左右翻頁與上下翻頁 |
| Interactive Postcard | Postcard.java  PostcardMailbox.java | 明信片互動元件 |
| Scroll Handler | ScrollHandler.java | 處理視窗scroll事件行為 |
| Interactive Scrollable | AutoScrollableLayout.java  AutoScrollableView.java  HorizonScrollableView.java  ScrollableImageHandler.java  ScrollableView.java  VerticalScrollableView.java | Scrollable互動元件 |
| Interactive Slideshow | SlideshowView.java  SlideshowViewActivity.java  SlideshowViewItem.java | Slideshow互動元件 |
| Interactive Video | MediaController.java  VideoPlayer.java | Video player互動元件 |
| Interactive Web View | InteractiveWebView.java  WebBrowser.java  WebBrowserActivity.java | Web view互動元件 |
| Interactive Youtube | DeveloperKey.java  YoutubeFragment.java  YouTubePlayActivity.java  YoutubeView.java | Youtube player互動元件 |

### Build Interactive Object Library

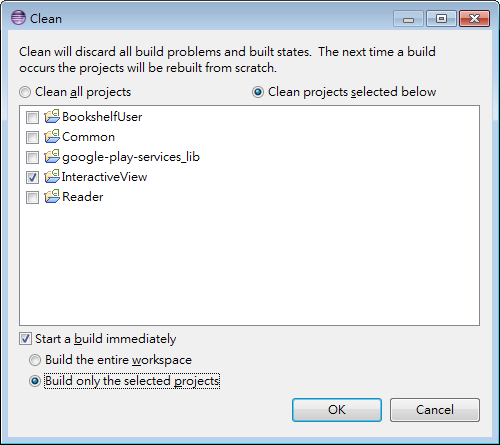
所有Interactive Object Library放在同一個Android Project，Project name是InteractiveView，目錄結構：



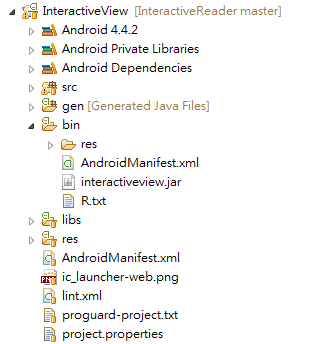
在Build Interactive Object Library時，記得要先確定Project properties內的”Is Library”要被選取



直接執行Eclipse的Clean and Build



Build complete後，會在bin資料夾產生interactiveview.jar即完成。



## Interactive Object Library Description

### Animation Fade

控制視窗切換時，使用淡入淡出來顯示切換。

|  |  |
| --- | --- |
| Function name | Description |
| void crossFade(final View showView, final View hideView, final Handler handler) | 將顯示與隱藏視窗傳入函數做視窗切換，handler是完成後接收message |

### Animation Rotate

控制視窗翻轉效果。

|  |  |
| --- | --- |
| Function name | Description |
| void applyRotation(View view, float start, float end, int nDirection) | 指定起始角度與結束角度來翻轉指定的視窗 |
| void setlistOnRotateEndListener(Rotate3d.OnRotateEndListener listener) | 設定call back，當翻轉結束會被通知 |

### Animation Move

控制視窗動畫移動。

|  |  |
| --- | --- |
| Function name | Description |
| void move(final View view, int nX1, int nX2, int nY1, int nY2, int nDuration, final Handler notifyHandler) | 指定起始的X Y位置與定位的X Y位置來移動指定的視窗，handler為完成後會收到完成的message通知 |

### Animation Zoom

控制視窗zoom in和zoom out動畫效果。

|  |  |
| --- | --- |
| Function name | Description |
| void zoom(View view, float fScaleSize) | 指定scale size去zoom指定的視窗，scale size小於1則zoom out，大於1則zoom in |

### Audio Player

互動元件Audio播放，包含播放控制列。

|  |  |
| --- | --- |
| Function name | Description |
| void init() | 新增播放元件與初始化屬性 |
| void setDisplay(int nX, int nY, int nWidth, int nHeight) | 設定視窗位置與大小 |
| void setPosition(int nChapter, int nPage) | 設定元件位於哪一章節和頁面 |
| void setPlayerControls(boolean bShow) | 是否顯示控制列 |
| void setOption(boolean bAutoPlay, boolean bAutoStop, boolean bLoop, int nStart, int nEnd) | 設定自動播放、自動停止、連續播放與播放起始點屬性 |
| void setImage(String strImagePath, int nWidth, int nHeight) | 設定視窗圖片 |
| void setAudioFile(String strFile) | 設定音樂檔路徑 |
| void play() | 播放音樂 |
| void stop() | 停播音樂 |
| void release() | 回收播放元件 |

### Interactive Button

按鈕互動元件。

|  |  |
| --- | --- |
| Function name | Description |
| void initButton() | 初始背景色、畫面屬性 |
| void setDisplay(int nX, int nY, int nWidth, int nHeight) | 設定視窗位置與大小 |
| void setPosition(int nChapter, int nPage) | 設定元件位於哪一章節和頁面 |
| void setImageSrc(String strSrc, String strTouchDown, String strTouchUp, int nWidth, int nHeight) | 設定視窗normal、down、up三種狀態圖片與大小。 |
| void setNotifyHandler(Handler handler) | 設定要接收通知的Handler |
| void setButtonClickType(int nClickType) | 設定按鈕點擊型態  BUTTON\_TYPE\_TAP：Click  BUTTON\_TYPE\_LONG\_PRESS：Long Click |
| void reset() | 回復按鈕一般狀態 |

按鈕的Event行為定義在onTouchListener內。當點擊時會發出通知給群組管理物件。

### Interactive Doodle

繪圖互動元件。

|  |  |
| --- | --- |
| Function name | Description |
| void setDisplay(int nX, int nY, int nWidth, int nHeight) | 設定視窗位置與大小 |
| void setPosition(int nChapter, int nPage) | 設定元件位於哪一章節和頁面 |
| void setImageSrc(String strImagePath, int nWidth, int nHeight) | 設定視窗圖片與大小。 |
| void setBrushes(int nBrushes) | 設定筆刷 |
| void setEraseWidth(int nWidth) | 設定橡皮擦寬度 |
| void setPalette(int nPalette) | 設定調色值 |
| void setEraserBtn(float nX, float nY, int nWidth, int nHeight, String strImagePath) | 設定橡皮擦位置大小與圖片 |
| void setPaletteBtn(float nX, float nY, int nWidth, int nHeight, String strImagePath) | 設定調色盤位置大小與圖片 |
| void setPenBtn(float nX, float nY, int nWidth, int nHeight, String strImagePath) | 設定繪筆位置大小與圖片 |
| void setResetBtn(float nX, float nY, int nWidth, int nHeight, String strImagePath) | 設定清除按鈕位置大小與圖片 |
| void setSaveBtn(float nX, float nY, int nWidth, int nHeight, String strImagePath) | 設定儲存按鈕位置大小與圖片 |

Doodle元件主要是透過FingerPaintView去做繪圖的處理，會新增一個FingerPainView在自己的視窗上面。

### Finger Paint

手繪功能視窗，視窗實做Draw繪圖功能。

|  |  |
| --- | --- |
| Function name | Description |
| void init() | 初始化筆刷、橡皮擦、畫布物件。 |
| void onDraw(Canvas canvas) | Call back function，當視窗呼叫invalidate函數時，會觸發onDraw，繪圖將在canvas物件上執行 |
| void touch\_start(float x, float y) | 繪圖起始點 |
| void touch\_move(float x, float y) | 手指移動繪圖，記錄移動路徑 |
| void touch\_up() | 手繪結束，保留手繪線條 |
| void processEvent(MotionEvent event) | 判斷手勢，執行手繪 |
| void clear() | 清除手繪畫面 |
| boolean isCapturing() | 是否執行手繪 |
| void setIsCapturing(boolean mCapturing) | 設定是否執行手繪 |
| void setEraser(boolean bEraser) | 設定是否執行橡皮擦功能 |
| void setBackground(String strImagePath) | 設定手繪視窗背景圖 |
| boolean exportBitmap(String strPath, int nWidth, int nHeight) | 將手繪視窗畫面匯出到指定的檔案路徑，匯出格式為bitmap物件 |
| void setTextToDraw(String strText) | 輸入文字到手繪視窗 |
| void setPenStrokeWidth(int nWidth) | 設定手繪線條寬度 |
| void setEraseStrokeWidth(int nWidth) | 設定橡皮擦寬度 |

### Flipper Show

控制多個視窗從上、下、左、右方向以滑動方式切換顯示。

|  |  |
| --- | --- |
| Function name | Description |
| void init(Context context) | 初始化Flipper物件與Animation屬性 |
| void setNotifyHandler(Handler handler) | 設定訊息通知接收端的Handler |
| int addChild(int nResId) | 新增視窗到Flipper Group，函數會透過Resource id將layout resource轉變為View物件，然後新增到Flipper內 |
| void showView(int nIndex) | 依據index顯示指定的視窗 |

Flipper show物件是一個視窗集合物件，控制多個視窗以滑動方式切換顯示。

### Gallery View

Gallery視窗物件，主要用於顯示書本所有頁面時用，主要是延伸Android HorizontalScrollView。

|  |  |
| --- | --- |
| Function name | Description |
| void init(Context context) | 初始化LinearLayout到HorizontalScrollView。 |
| void setCurrentItem(int nItem) | 將視窗scroll到指定的item位置 |
| void setOnScrollStopListner(GalleryView.onScrollStopListner listner) | 設定Scroll停止時的call back |
| void setItem(SparseArray<SparseArray<String>> listImage, int nWidth, int nHeight) | 新增view item到Gallery視窗 |
| void setViewCenter() | Scroll view to center |
| void scrollToStart() | Scroll到最左邊 |
| void scrollToEnd() | Scroll到最右邊 |