# AppCross Common Library Specification

Version 0.0.1

內容

[AppCross Common Library Specification 1](#_Toc385262200)

[Version 0.0.1 1](#_Toc385262201)

[1. 概述 3](#_Toc385262202)

[2. Common Class 3](#_Toc385262203)

[2.1 Catalog 3](#_Toc385262204)

[2.2 Build Common Library 5](#_Toc385262205)

[3. Common Library Description 8](#_Toc385262206)

[3.1 Bitmap Handler 8](#_Toc385262207)

[3.2 Clear Cache 9](#_Toc385262208)

[3.3 XML Parser 9](#_Toc385262209)

[3.4 Device 10](#_Toc385262210)

[3.5 Event Handler 10](#_Toc385262211)

[3.6 Event Message 10](#_Toc385262212)

[3.7 File Handler 12](#_Toc385262213)

[3.8 Sqlite Handler 13](#_Toc385262214)

[3.9 Thread Handler 14](#_Toc385262215)

[3.10 Data Type 14](#_Toc385262216)

[3.11 Version Handler 15](#_Toc385262217)

[3.12 Word Handler 15](#_Toc385262218)

[3.13 Zip Handler 15](#_Toc385262219)

## 概述

AppCross Common Library 主要提供Android AppCross Reader程式共用的函數與物件，在設計上會將每個功能設計成獨立的模組，共用元件包含了Zip、Device、Bitmap、File、Event….等行為的功能。

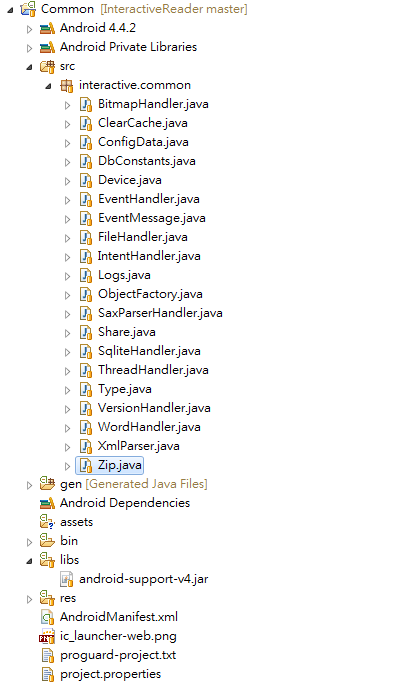
## Common Class

### Catalog

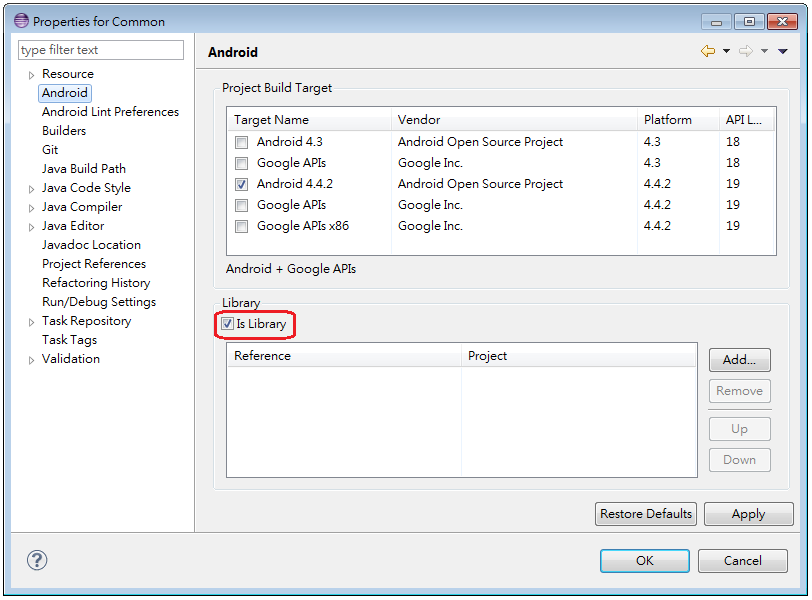
|  |  |  |
| --- | --- | --- |
| **Module Name** | **File** | **Description** |
| Bitmap Handler | BitmapHandler.java | Handle bitmap resize、read bitmap、rotate、combine、cut |
| Clear Cache | ClearCache.java | Clear android process cache file |
| XML Parser | XmlParser.java  SaxParserHandler.java  ConfigData.java | Parse XML format file |
| Device | Device.java | Get scale size、device size、android version、display size。 |
| Event Handler | EventHandler.java | Send message to handler |
| Event Message | EventMessage.java | Define message to send |
| File Handler | FileHandler.java | Read file、find file |
| Intent Handler | IntentHandler.java | Handle Intent result |
| Debug Log | Logs.java | Show debug log |
| Image button handler | ObjectFactory.java | Handle Image button |
| Share | Share.java | Call android share intent |
| Sqlite Handler | SqliteHandler.java | Handle SQLite DB object |
| Thread Handler | ThreadHandler.java | Create and Handle thread |
| Data type | Type.java | Define data type |
| Version Handler | VersionHandler.java | Get android version name |
| Word Handler | WordHandler.java | Get string from android string.xml |
| Zip Handler | Zip.java | Express zip file |

### Build Common Library

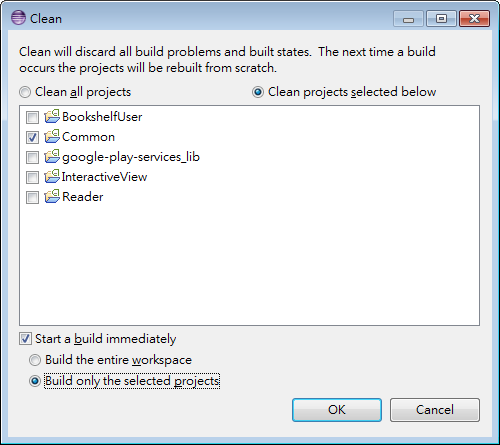
所有Common Library放在同一個Android Project，Project name是Common，目錄結構：



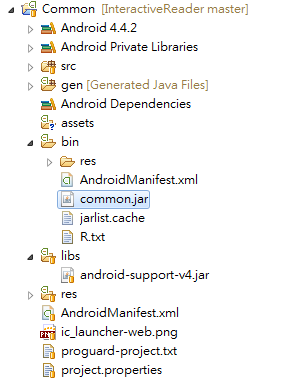
在Build Common Library時，記得要先確定Project properties內的”Is Library”要被選取



直接執行Eclipse的Clean and Build



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



## Common Library Description

### Bitmap Handler

* 主要處理程式UI圖檔跟動態Load書籍圖檔，由於Android的cache size不大，所以Bitmap Handler在load bitmap時會採用FileInputStream去讀取bitmap檔減少使用到Android的cache。
* Bitmap Handler在讀取圖檔時亦會偵測圖檔size，如果圖檔size大於4096\*4096則會自動縮小圖檔size為4096\*4096。
* Bitmap Handler提供combine bitmap跟cut bitmap函數，主要處理互動元件的scrollable功能。
* loadBitmapFromView函數則是抓取Device 上view的畫面，將畫面存成bitmap。

|  |  |
| --- | --- |
| Function name | Description |
| int getMaxTexDim() | 讀取Android最大支援的解圖Size |
| Bitmap readBitmap(Context context, int resId) | 從resouce id讀取圖檔 |
| Bitmap readBitmap(Context context, String strFilePath, int reqWidth, int reqHeight, boolean bResize) | 從圖檔路徑讀取圖檔，將圖檔依據指定的size輸出bitmap物件，會先清除cache file |
| Bitmap readBitmap(String strFilePath, int reqWidth, int reqHeight, boolean bResize) | 從圖檔路徑讀取圖檔，將圖檔依據指定的size輸出bitmap物件，不會先清除cache file |
| int calculateInSampleSize(BitmapFactory.Options options, int reqWidth, int reqHeight) | 計算圖檔resize時需要的scale size |
| Bitmap getResizedBitmap(Bitmap bm, int newWidth, int newHeight) | Resize bitmap |
| Bitmap combineBitmap(Bitmap background, Bitmap foreground, float left, float top) | 兩張圖檔合併 |
| Bitmap getScreenshotsForCurrentWindow(Activity activity) | 快照畫面用bitmap物件輸出 |
| Bitmap cutBitmap(Bitmap mBitmap, Rect r, Bitmap.Config config) | 切割圖檔 |
| int getBitmapWidth(String strFilePath) | 取得圖檔寬度 |
| int getBitmapHeight(String strFilePath) | 取得圖檔高度 |
| Bitmap loadBitmapFromView(Context context, View v) | 將View物件畫面輸出成bitmap物件 |
| void releaseBitmap(Bitmap bitmap) | Recycle bitmap object |

### Clear Cache

清理AppCross Reader程式執行時產生的cache files。

|  |  |
| --- | --- |
| Function name | Description |
| void clearApplicationData(Context context) | 清除程式暫存檔與cache file |
| void trimCache(Context context) | 清除程式cache |

### XML Parser

讀取書籍的definition.xml將資料存到configData資料結構內，XML Parser採用SAX object去parse XML檔。

|  |  |
| --- | --- |
| Function name | Description |
| void parse(File file, ConfigData configData) | Parse XML file將資料存到ConfigData資料結構 |

### Device

抓取Device display size，scale size、Android SDK Version。

|  |  |
| --- | --- |
| Function name | Description |
| int getDisplayWidth() | 取得螢幕寬長的解析度 |
| int getDisplayHeight() | 取得螢幕高長的解析度 |
| int getDeviceWidth() | 取得螢幕實際寬度 |
| int getDeviceHeight() | 取得螢幕實際高度 |
| float getScaleSize() | 根據解析度與實際寬高計算scale size |
| int getOrientation() | 取的Device目前的直橫狀態 |
| boolean isLargeDisplaySize() | 判斷螢幕是否為大size螢幕 |
| double getDisplayIncheSize() | 取得螢幕inch size |
| int getDeviceType() | 判斷螢幕是phone或是table |
| boolean checkInstallation(String packageName) | 判断是否安装指定的應用程式 |
| void installApp(Activity activity, String strAppPackageName) | 引導跳轉去Google Play上某個應用的詳細頁面 |
| int getAndroidSdkVersion() | 取得Android SDK version |
| String getAndroidReleaseVersion() | 取得Android API version |

### Event Handler

透過notify函數傳送Message給特定的Handler。

|  |  |
| --- | --- |
| Function name | Description |
| void notify(Handler handler, int nWhat, int nEvent, int nPosition, Object object) | 傳送Message到指定的Handler |

### Event Message

定義程式所有的Message ID，從1024開始累加。

**public** **static** **final** **int** *MSG\_CUSTOM* = 1024;

**public** **static** **final** **int** *MSG\_CHAPTER* = *MSG\_CUSTOM* + 1;

**public** **static** **final** **int** *MSG\_PAGE* = *MSG\_CUSTOM* + 2;

**public** **static** **final** **int** *MSG\_VIEW\_INITED* = *MSG\_CUSTOM* + 3;

**public** **static** **final** **int** *MSG\_WEB* = *MSG\_CUSTOM* + 4;

**public** **static** **final** **int** *MSG\_JUMP* = *MSG\_CUSTOM* + 5;

**public** **static** **final** **int** *MSG\_VIEW\_CHANGE* = *MSG\_CUSTOM* + 6;

**public** **static** **final** **int** *MSG\_MEDIA\_PLAY* = *MSG\_CUSTOM* + 7;

**public** **static** **final** **int** *MSG\_MEDIA\_PAUSE* = *MSG\_CUSTOM* + 8;

**public** **static** **final** **int** *MSG\_MEDIA\_STOP* = *MSG\_CUSTOM* + 9;

**public** **static** **final** **int** *MSG\_SHOW\_PROGRESS* = *MSG\_CUSTOM* + 10;

**public** **static** **final** **int** *MSG\_FLIPPER\_CLOSE* = *MSG\_CUSTOM* + 11;

**public** **static** **final** **int** *MSG\_NOTIFY\_ROTATE* = *MSG\_CUSTOM* + 12;

**public** **static** **final** **int** *MSG\_IMAGE\_CLICK* = *MSG\_CUSTOM* + 13;

**public** **static** **final** **int** *MSG\_SHOW\_ITEM* = *MSG\_CUSTOM* + 14;

**public** **static** **final** **int** *MSG\_DOUBLE\_CLICK* = *MSG\_CUSTOM* + 15;

**public** **static** **final** **int** *MSG\_START\_UNEXPRESS* = *MSG\_CUSTOM* + 16;

**public** **static** **final** **int** *MSG\_CHECKED\_BOOK* = *MSG\_CUSTOM* + 17;

**public** **static** **final** **int** *MSG\_GO\_FORWARD* = *MSG\_CUSTOM* + 18;

**public** **static** **final** **int** *MSG\_OPTION\_ITEM\_SELECTED* = *MSG\_CUSTOM* + 19;

**public** **static** **final** **int** *MSG\_LOCK\_PAGE* = *MSG\_CUSTOM* + 20;

**public** **static** **final** **int** *MSG\_UNLOCK\_PAGE* = *MSG\_CUSTOM* + 21;

**public** **static** **final** **int** *MSG\_LOCK\_HORIZON* = *MSG\_CUSTOM* + 22;

**public** **static** **final** **int** *MSG\_UNLOCK\_HORIZON* = *MSG\_CUSTOM* + 23;

**public** **static** **final** **int** *MSG\_LOCK\_VERTICAL* = *MSG\_CUSTOM* + 24;

**public** **static** **final** **int** *MSG\_UNLOCK\_VERTICAL* = *MSG\_CUSTOM* + 25;

**public** **static** **final** **int** *MSG\_ACTIVITY\_RESULT* = *MSG\_CUSTOM* + 26;

**public** **static** **final** **int** *MSG\_ANIMATION\_END* = *MSG\_CUSTOM* + 27;

**public** **static** **final** **int** *MSG\_DRAG\_END* = *MSG\_CUSTOM* + 28;

**public** **static** **final** **int** *MSG\_CURRENT\_ACTIVE* = *MSG\_CUSTOM* + 29;

**public** **static** **final** **int** *MSG\_BROWSER\_CLOSE* = *MSG\_CUSTOM* + 30;

**public** **static** **final** **int** *MSG\_DRAG\_START* = *MSG\_CUSTOM* + 31;

**public** **static** **final** **int** *MSG\_SEND\_POSTCARD* = *MSG\_CUSTOM* + 32;

**public** **static** **final** **int** *MSG\_NOT\_CURRENT\_ACTIVE* = *MSG\_CUSTOM* + 33;

**public** **static** **final** **int** *MSG\_BUTTON\_EVENT* = *MSG\_CUSTOM* + 34;

**public** **static** **final** **int** *MSG\_RESET* = *MSG\_CUSTOM* + 35;

**public** **static** **final** **int** *MSG\_ANIMATION\_ZOOM* = *MSG\_CUSTOM* + 36;

**public** **static** **final** **int** *MSG\_ANIMATION\_MOVE* = *MSG\_CUSTOM* + 37;

**public** **static** **final** **int** *MSG\_JUMP\_FADE* = *MSG\_CUSTOM* + 38;

**public** **static** **final** **int** *MSG\_GCM\_REGISTERED* = *MSG\_CUSTOM* + 39;

**public** **static** **final** **int** *MSG\_LOGIN* = *MSG\_CUSTOM* + 40;

**public** **static** **final** **int** *KEY\_BACK* = 4;

### File Handler

讀取指定的檔案，搜尋檔案，主要用在AppCross Reader尋找書籍檔時用到。

|  |  |
| --- | --- |
| Function name | Description |
| boolean unExpressFile(final String strSrcPath, final String strDistPath) | 解壓縮檔案，會呼叫zip模組去解壓縮 |
| void createPath(final String strPath) | 建立檔案路徑 |
| String getStringFromFile(String filePath) | 讀取檔案內容 |
| String convertStreamToString(InputStream is) | 將stream輸出成String物件 |
| boolean checkAssetFile(Activity activity, String strFile) | 檢查assets資料夾是否有指定的檔案存在 |
| void copyFileFromAssets(Activity activity, String file, String dest) | 將assets內指定的檔案複製到指定的資料夾路徑 |
| int getBookInfo(Activity activity, SparseArray<String> saResInfo) | 讀取assets內resource檔的內容 |
| String searchBookFile(String strBookFolder, String strFindFile) | 到指定的路徑搜尋指定的檔案 |
| String getSdcardPath() | 取得系統SD card路徑 |
| String searchFilePath(String path, String find) | 取得指定檔案的路徑 |
| String getObbFile(String strPath, String strPackName) | 搜尋OBB內是否有指定的檔案 |
| boolean fileRename(String strOldPath, String strOldFile, String strNewPath, String strNewFile) | 更改檔案路徑與檔名 |
| boolean fileCopy(String strOldPath, String strOldFile, String strNewPath, String strNewFile) | 複製檔案到指定的資料夾 |
| boolean deleteFile(String strFile) | 刪除檔案 |
| boolean delete(String strFile) | 刪除檔案與路徑 |
| StringBuffer getFileContent(String path) | 讀取檔案內容 |
| boolean isFileExist(String strFilePath) | 判斷檔案是否存在 |
| boolean checkPath(String strPath, boolean bCreate) | 判斷路徑是否存在 |

Intent Handler

主要用在Device照相功能，與Device Camera Intent溝通。

|  |  |
| --- | --- |
| Function name | Description |
| Bitmap activityResult(Activity activity, int requestCode, int resultCode, Intent data) | Android Camera intent call back，  Activity 的onResult會被呼叫，呼叫此函數抓取照片 |

Debug Log

統一的除錯諮詢，程式透過ShowTrace顯示除錯訊息。

|  |  |
| --- | --- |
| Function name | Description |
| void showTrace(String msg) | 顯示Log於Android Logcat view |
| void complain(Context context, String message) | 顯示Android Alert Dialog |

Image Button Handler

處理Image Button三種狀態的變化(Down,Up,Normal)。

|  |  |
| --- | --- |
| Function name | Description |
| ImageView getImageButton(int nId) | 從resource id取得image view物件 |
| int addImageButton(int nId, int nNormalId, int nTouchDownId, int nTouchUpId) | 新增image view行為圖檔id |
| void setImgBtnTouchDown(int nId) | 設定image view touch down時的圖檔 |
| void setImgBtnTouchUp(int nId) | 設定image view touch up時的圖檔 |
| void setImgBtnNormal(int nId) | 設定image view touch normal時的圖檔 |

Share

Android 分享Intent，會啟動Android share dialog。

|  |  |
| --- | --- |
| Function name | Description |
| void shareAll(String strTitle, String strSubject, String strMessage, SparseArray<String> listImagePath) | 呼叫Android share intent |

### Sqlite Handler

Create SQLite DB新增Table新增預設的資料，提供讀取與寫入的介面。

|  |  |
| --- | --- |
| Function name | Description |
| void onCreate(SQLiteDatabase db) | Call back function 如果DB不存在則create DB |
| void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) | Call back function版本更新時被呼叫 oldVersion=舊的資料庫版本；newVersion=新的資料庫版本 |
| void onOpen(SQLiteDatabase db) | Call back function每次成功打開數據庫後首先被執行 |
| void close() | 關閉資料庫 |
| void addFavorite(int nChapter, int nPage) | 加入reader我的最愛資料到資料庫 |
| void deleteFavorite(int nChapter, int nPage) | 從資料庫刪除我的最愛資料 |
| void updateFavorite(int nOldChapter, int nOldPage, int nNewChapter, int nNewPage) | 更新資料庫我的最愛資料 |
| void getFavoriteData(SparseArray<FavoriteData> listData) | 取得我的最愛資料 |

### Thread Handler

Create Thread 設定thread屬性與執行。

|  |  |
| --- | --- |
| Function name | Description |
| void setThdId(int nThdId) | 設定thread id |
| void start() | 開始執行thread |
| void setName(String name) | 設定thread名稱 |
| void join() | Thread join |
| void setPriority(int androidOsPriority) | 設定thread屬性 |

### Data Type

定義基本資料型態。

**public** **static** **final** **int** *TRUE* = 1;

**public** **static** **final** **int** *FALSE* = 0;

**public** **static** **final** **int** *INVALID* = -1;

**public** **static** **final** **int** *VALID* = *TRUE*;

**public** **static** **final** **int** *TRACE\_LEVEL\_SIMPLE* = 1;

**public** **static** **final** **int** *TRACE\_LEVEL\_NORMAL* = *TRACE\_LEVEL\_SIMPLE* + 1;

**public** **static** **final** **int** *TRACE\_LEVEL\_DETAIL* = *TRACE\_LEVEL\_NORMAL* + 1;

**public** **static** **final** **int** *SMALLEST\_SCREEN\_WIDTH\_DP* = 600;

**public** **static** **final** **int** *DEVICE\_PHONE* = 0;

**public** **static** **final** **int** *DEVICE\_TABLET* = 1;

**public** **static** **final** String *DEFAULT\_STORAGE* = "/sdcard/download/";

### Version Handler

抓取Android SDK Version code和Version name。

|  |  |
| --- | --- |
| Function name | Description |
| int getVersionCode(Context context) | 取得Android SDK Version code |
| String getVersionName(Context context) | 取得Android API Version name |

### Word Handler

抓取Android文字定義檔string.xml內的定義文字。

|  |  |
| --- | --- |
| Function name | Description |
| String getString(Context context, String strName) | 取得android string.xml描述字 |

### Zip Handler

Zip檔解壓縮模組。

|  |  |
| --- | --- |
| Function name | Description |
| void zip(String[] files, String zipFile) | 壓縮檔案 |
| boolean unzip(String zipFile, String location) | 解壓縮檔案 |