

# TizzBird Customization Guide

## Smart Media Player **TIZZBIRD™ F Series**



**Rev 1.03**

*For firmware Rev 1.23 or later*

**Table of Contents**

- 1. Overview ..... 3**
- 2. Customize User Interface ..... 4**
  - 2.1 Customize Background Screen ..... 4**
  - 2.2 Customize Home Screen Items ..... 4**
  - 2.3 Make it Permanent Change and Restore Default ..... 5**
- 3. Auto Playback ..... 6**
  - 3.1 Autorun function ..... 6**
  - 3.2 Auto folder playback ..... 6**
  - 3.3 Auto file playback ..... 7**
- 4. Auto Installation & Execution of APPs ..... 8**
  - 4.1 Auto Installation ..... 8**
  - 4.2 Auto Execution of Pre-installed APP ..... 9**
  - 4.3 Auto Installation and Execution ..... 10**

## 1. Overview

TizzBird is a quite flexible and powerful device, and it can be customizable in many ways. Besides just AUTORUN or AUTO CONFIGURATION, you can pre-install some applications and run it directly without intervention of TizzBird UI.

You need the customizations of TizzBird at following cases:

- *Need to change the background and item images.*
- *Need to playback automatically some images or videos.*
- *Need to execute some app that was pre-installed.*
- *Need to install customized application and execute that app every booting time.*

These functions let wide-open the business chances for VAR (value added reseller) and small companies who want to advertise their video or run their own app.

### ※ **Caution & Important Notice**

*All of the customization start with "[MMP\\_Auto\\_Config.xml](#)" file.*

This is quite a simple text XML file, so you can easily create or edit with simple text editor. This file should reside at the root folder of drive.

For example, put this MMP\_Auto\_Config.xml at the USB flash drive, reboot, then the contents of this XML file will be executed.

Common header of this XML file is like this:

```
<?xml version="1.0" encoding="utf-8"?>
<AndroidConfig Operation="AutoConfig"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
```

[This common header should exist on every "MMP\\_Config\\_Auto.xml" file.](#)

## 2. Customize User Interface

### 2.1 Customize Background Screen

You can easily change the background (or wallpaper) of home screen with following script at "MMP\_Auto\_Config.xml" file.

```
<HomeConfig>
    <Wallpaper image="ImageName" keepAspectRatio="true" />
</HomeConfig>
```

Just put the "ImageName" field to any image you want to replace. We recommend the background image size 1280x720 or 1920x1080 which fits the aspect ratio.

### 2.2 Customize Home Screen Items

You can easily change the image of each item of home screen easily with following script at "MMP\_Auto\_Config.xml" file. We recommend the image size of item would be 640x480 pixels.

```
<HomeConfig>
    <Item image="ImageName" name="ShowingTitleOfItem",
        action="ActionName" />
</HomeConfig>
```

Or you can configure the "Android Package" name and "Activity" name as well to re-direct the action like this:

```
<HomeConfig>
    <Item image="ImageName" name="ShowingTitleOfItem",
        package_name="PackageName",
        activity_name="AcitivityName"/>
</HomeConfig>
```

For example if you want to just change images of the current home screen, please make script like below:

```
<HomeConfig>
  <Item image="/home_file.jpg" name="File"
  action="com.android.mediacore.FILE.BROWSER" />
  <Item image="/home_photo.jpg" name="Photo"
  action="com.android.mediacore.PHOTO.BROWSER" />
  <Item image="/home_music.jpg" name="Music"
  action="com.android.mediacore.MUSIC.BROWSER" />
  <Item image="/home_movie.jpg" name="Movie"
  action="com.android.mediacore.MOVIE.BROWSER" />
  <Item image="/home_settings.jpg" name="Settings"
  action="com.android.mediacore.SETTINGS.BROWSER" />
  <Item image="/home_app.jpg" name="Application"
  action="com.android.mediacore.APPLICATION.BROWSER" />
  <Item image="/home_web.jpg" name="Web Browser"
  package_name="com.android.browser"
  activity_name="com.android.browser.BrowserActivity" />
</HomeConfig>
```

Place all of the images like above name and put them on the root folder of drive.

## 2.3 Make it Permanent Change and Restore Default

You can easily make the change permanent or restore to default image with following script.  
To make the change permanent (the changed images will come-up whenever boot-up):

```
<HomeConfig>
  <Config backup="true" />
</HomeConfig>
```

To restore the default TizzBird image:

```
<HomeConfig>
  <clear="true" />
</HomeConfig>
```

## 3. Auto Playback

### 3.1 Autorun function

You can make the "autorun" function just like other media player. To playback all the files in the "Autorun" folder, please copy the following "MMP\_Auto\_Config.xml" file on the root then, follow the procedures like below:

```
<?xml version="1.0" encoding="utf-8"?>
<AndroidConfig Operation="AutoConfig"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">

    <AutoPlay count="0">
        <Folder name="/Autorun" type="1"/>
    </AutoPlay>
</AndroidConfig>
```

- 1) Copy the "MMP\_Auto\_Config.xml" file on the root folder.
- 2) Create new folder name "Autorun".
- 3) Copy all the video files you want to play into the "Autorun" folder.

### 3.2 Auto folder playback

To playback some folder, write the script like this:

```
<AutoPlay count=n>
    <Folder name="folderName" type="1"/>
</AutoPlay>
```

You can configure media type at "type" field

**1=Movie, 2= Music, 3=Photo + background music**

You can also configure the repeat count on the "count=*number*" field.

**0=infinite repeat,**

**1..n=repeat as designated count**

### 3.3 Auto file playback

To designate automatic playback of some files, please write the script like this:

```
<AutoPlay count=n>  
  <File name="fileName1" />  
  <File name="fileName2" />  
</AutoPlay>
```

## 4. Auto Installation & Execution of APPs

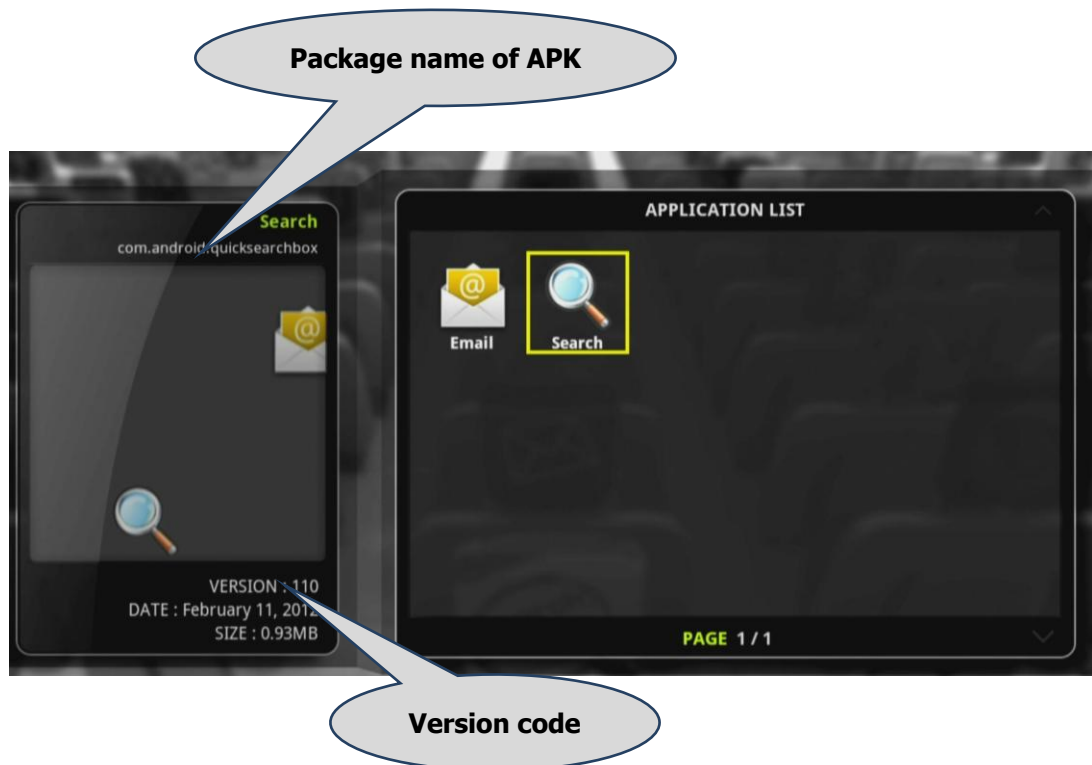
### 4.1 Auto Installation

Before proceed to install the APP automatically at TizzBird device, you need to know the following information about APP.

APK name: package name of the APK

Version Code: version code of APK file

The above information can be found on TizzBird APP browser as follows:



After getting two fields value, write the script as follows:

```
<AutoInstall>
  <APK name="packageName" version_code="versionCode"
    target_device="0" file="apkFileName" />
</AutoInstall>
```

The "target\_device" field designates the storage where APP should install.

**0=internal NAND, 1=SD card (if exists)**

The "file" field designates the location of APK file resides.

For above example, your script should be



```
<AutoInstall>
  <APK name="com.android.quickSearchBox"
    version_code="110" target_device="0"
    file="/search.APK" />
</AutoInstall>
```

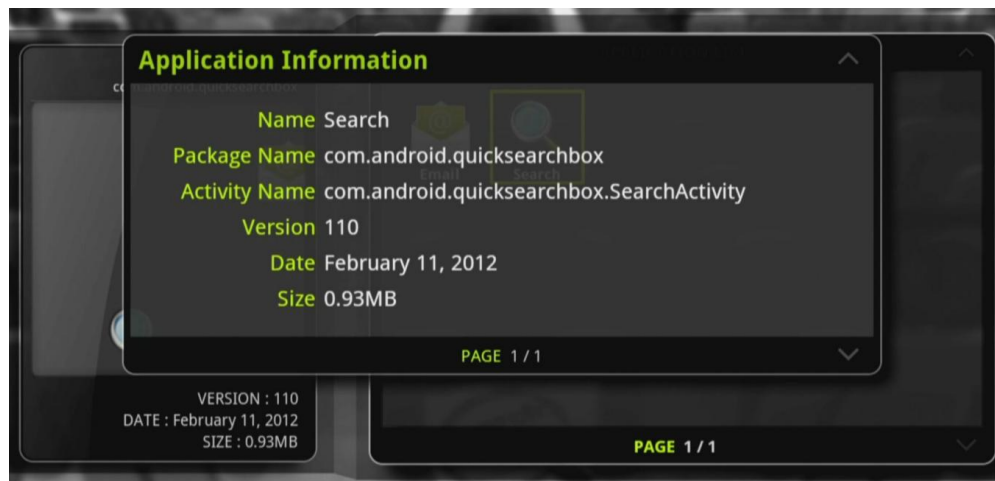
Here's another example for install "AngryBird" APP into TizzBird internal NAND.

```
<AutoInstall>
  <APK name=" com.rovio.angrybirdsseasons"
    version_code="142" target_device="0"
    file="/AngryBirds1.5.1.1.apk"/>
</AutoInstall>
```

## 4.2 Auto Execution of Pre-installed APP

If you have already installed the APP, and there can be a need to execute the APP directly without intervention of TizzBird U.I.

In that case you need to know the "**Package Name**" and "**Activity Name**" of the APP to execute. You can get those names at TizzBird APP browser by **typing "1234"**. The pop-up screen will come out as follows:



After getting two fields value, write the script as follows:

```
<AutoRun>
  <APK name="packageName" activity="activityName"
    cmdURL="optionalURL-Link"
    register="false or true" />
</AutoRun>
```

“**register**” field designate whether it should run or not after removing source drive.

true= the installed application run whenever reboot.

false= the installed application will not run when reboot.

For example, you can run the WEB browser directly as following scripts:

```
<AutoRun>
  <APK name="com.android.browser"
    activity="com.android.browser.BrowserActivity"
    cmdURL="http://www.naver.com" />
</AutoRun>
```

### 4.3 Auto Installation and Execution

You can also combine the Installation and Executing APP as following script:

```
<AutoInstall>
  <APK name="packageName" version_code="versionCode"
    target_device="0" file="apkFileName" />
</AutoInstall>
<AutoRun>
  <APK name="packageName"
    activity="activityName"
    cmdURL="optionalURL-Link" />
</AutoRun>
```

For example if you want to execute the AngryBirds with installation, the script should be:

```
<AutoInstall>
  <APK name=" com.rovio.angrybirds"
    version_code="142" target_device="0"
    file="/AngryBirds1.5.1.1.apk"/>
</AutoInstall>
<AutoRun>
  <APK name="com.rovio.angrybirds"
    activity="com.rovio.ka3d.App"
</AutoRun>
```