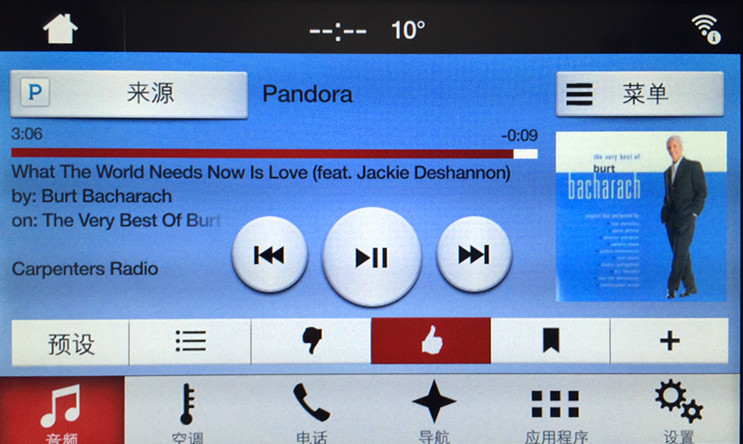
# Applink应用在Gen3 图片的支持

## Gen3新特性

Ford Gen3支持8寸彩色显示屏，在Applink这块，我们添加了对图片、进度条的支持，使得应用在车机上的展现更加丰富多彩。  
这是已经上线的Pandora 在Sync3 上的展现形式：



以下是我们一个典型的音乐类Applink应用在Gen3上的展现形式：



相对于之前的Gen1.1, Gen3增加了以下元素：

1. 应用图标

应用可以上传自己的图标到车机上，图标会在移动应用程序列表和 音源地方展示

1. 进度条

应用可以根据音乐长度设置进度条

1. 专辑图片

应用可以将专辑图片显示到车机上

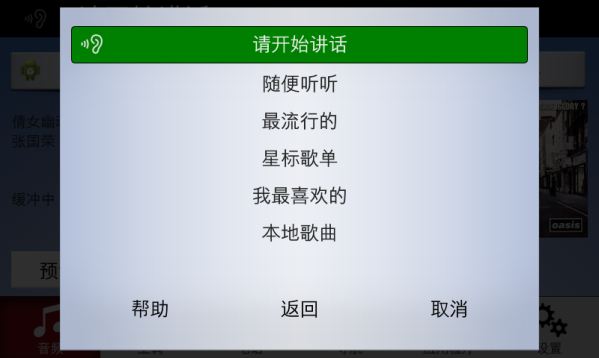
1. 触摸屏按键

Gen3上没有实体的OK键，该部分都在触摸屏上实现

1. Softbutton

可以定制SoftButton上的图片，图片格式以及大小见附件。之前的播放/暂停 图片已经不再支持

1. 设置帮助列表

可以定制用户说出名字后的语音提示列表，如下图,

## 2.适配Gen3主要改动

对于大部分音乐类的应用，为适配Gen3 需要做以下更改：

1. 添加应用图标
2. 增加音乐进度条显示
3. 上传专辑图片
4. 如果之前没有定义OK键，添加OK键的播放/暂停功能
5. 修改之前的SoftButton，一般指带有播放/暂停功能的Softbutton
6. 设置帮助列表，可以定制用户说出名字后的语音提示列表

在后面一段时间 Gen1.1 和Gen3会一直在市场上，所以Applink应用还需要兼容Gen1.1 和 Gen3。因此还需要根据App连接上的车机类型等来判断是否是Gen3 并做图片相关的RPC。

## 3.参考代码

针对以上 的六个改动，参考代码如下：

### 判断车机是否是支持图片显示

- (void)onRegisterAppInterfaceResponse:(FMCRegisterAppInterfaceResponse \*)response

{

self.isGraphicsSupported = NO; //Variable to store if Graphics are supported

// Get the display capabilities

if (response.displayCapabilities != nil) {

self.displayType = response.displayCapabilities.displayType;

self.textFields = response.displayCapabilities.textFields;

self.templatesAvailable = response.displayCapabilities.templatesAvailable;

if (response.displayCapabilities.graphicSupported != nil) {

self.isGraphicsSupported = [response.displayCapabilities.graphicSupported boolValue];

}

}

}

***Header:***

@property (strong, nonatomic) FMCDisplayType\* displayType;

@property (strong, nonatomic) NSArray\* textFields;

@property (strong, nonatomic) NSArray\* templatesAvailable;

@property (assign) BOOL isGraphicsSupported;

// WHEN YOU START THE PROXY SET firstHmiNone to true

firstHmiNone **=** **true;**

public void onOnHMIStatus**(**OnHMIStatus notification**)** **{**

**[...]**

**case** HMI\_NONE**:** **{**

Log**.**i**(**TAG**,** "HMI\_NONE"**);**

**if** **(**firstHmiNone**)** **{**

isGraphicsSupported **=** **false;**

displayType **=** proxy**.**getDisplayCapabilities**().**getDisplayType**();**

textFields **=** proxy**.**getDisplayCapabilities**().**getTextFields**();**

templatesAvailable **=** proxy**.**getDisplayCapabilities**().**getTemplatesAvailable**();**

**if** **(**proxy**.**getDisplayCapabilities**().**getGraphicSupported**()** **!=** **null)** **{**

isGraphicsSupported **=** proxy**.**getDisplayCapabilities**().**getGraphicSupported**();**

**}**

// Check the DisplayType for Gen3

firstHmiNone **=** **false;**

**}**

**break;**

**}**

**}**

***Local Fields:***

private boolean firstHmiNone**;**

private boolean isGraphicsSupported**;**

private DisplayType displayType**;**

private Vector**<**TextField**>** textFields**;**

private Vector**<**String**>** templatesAvailable**;**

在Gen1.1 上不支持图片，只有在Gen3和未来的机型上支持图片。在添加应用图标之前我们需要判断 车机是否支持图片。这就可以利用 *判断车机是否是支持图片显示* 中的变量 isGraphicsSupported. 如果该值是true，我们才需要做Gen3所需改动。

Gen3上Applink图标支持png,bmp,jpg 三种格式，要在车机上显示图片需要有两个步骤：

1. 利用putfile 这个API将图片上传到车机上
2. 利用Show SetAppIcon 等api显示图片

### 显示应用图标

显示应用图标分为两个步骤，在收到第一个HMI STATUS时先上传应用图标到车机上，在上传成功后调用setAppIcon api设置应用图标。

#### 上传图片到车机

目前Gen3支持png jpg bmp格式图片，appicon的大小为70X70 px。上传图片到车机的API为PutFile，其中两个参数最为重要：

1. syncFileName 作为图片在SYNC中的名称
2. fileType 图片的类型(png bmp jpg)

其他的参数可以设置成Nil或0，下面的示例代码展示了如何上传图片到车机。另外，在实际代码中需要保存correlationID以便在上传完成后 设置该图片为应用图标。

iOS code：

// Get the image data

UIImage\* pngImage = [UIImage imageNamed: @"icon.png"]; //Load an Image

NSData\* pngData = UIImagePNGRepresentation(pngImage); //Save as raw Data

// Upload the file

FMCFileType\* fileType = [FMCFileType GRAPHIC\_PNG]; //Save Filetype

FMCPutFile\* putRequest = [[FMCPutFile alloc] init]; //Initiate putFile RPC Request

[putRequest setCorrelationID:[NSNumber numberWithInt:autoIncCorrID++]];

[putRequest setFileType:fileType];

[putRequest setPersistentFile:[NSNumber numberWithBool:NO]]; //Can be disregarded

[putRequest setSyncFileName:@"icon.png"]; //Handle that will be used to call the image

[putRequest setSystemFile:[NSNumber numberWithBool:NO]]; //Disregarded

[putRequest setOffset:[NSNumber numberWithInt:0]]; //Disregarded

[putRequest setLength:[NSNumber numberWithInt:0]]; //Disregarded

[putRequest setBulkData:pngData]; //Asign the raw data to the rpc request

[proxy sendRPCRequest:putRequest]; //Send out request

Android Code:

//创建PutFile和设置参数

PutFile msg **=** **new** PutFile**();**

msg**.**setSyncFileName**(**“AppIcon.png”**);**

msg**.**setFileType(FileType.GRAPHIC\_PNG);

self.appIconCorrID = autoIncCorrId; //Save for later usage

msg**.**setCorrelationID**(**autoIncCorrId**++);**

//图片转化为为byte数组上传到SYNC

//The BitmapFactory.decodeResource() will return a decoded bitmap, or null if the image could not be decoded.

//The R.drawable.icon reference is the physical graphic (in this case the file is named icon.png) located in the projects resources drawable directory

//See http://developer.android.com/reference/android/graphics/BitmapFactory.html for further details.

Bitmap photo **=** BitmapFactory**.**decodeResource**(**getResources**(),** R**.**drawable**.**icon**);**

//A specialized OutputStream for class for writing content to an (internal) byte array.

//See http://developer.android.com/reference/java/io/ByteArrayOutputStream.html for further details.

ByteArrayOutputStream bas **=** **new** ByteArrayOutputStream**();**

photo**.**compress**(**CompressFormat**.**PNG**,** 100**,** bas**);**

byte**[]** data **=** **new** byte**[**bas**.**toByteArray**().**length**];**

data **=** bas**.**toByteArray**();**

//将图片数据设置到PutFile中

msg**.**setBulkData**(**data**);**

**try**

**{**

//send the PutFile RPC request off to SYNC

proxy**.**sendRPCRequest**(**msg**);**

**}**

**catch** **(**SyncException e**)**

**{**

//if an exception was encountered, log it to the Android Logcat and handle the exception

e**.**printStackTrace**();**

Log**.**e**(**TAG**,** "Exception encountered when trying to send RPC request to SYNC: " **+** e**.**toString**());**

**}**

#### 设置应用程序图标

为了使用户在应用连接的时候尽早看到应用图标，一般在SYNC找到应用后 就开始上传图片和设置图标。 对于iOS，需要在registerAppInterfaceResponse中上传，对于Android，则是在收到第一个HMI\_STATUS时候上传。首先调用putFile将图片上传到SYNC，在PutFileResponse中 一旦上传成功，就是用setAppIcon来设置程序图标。

iOS code: ***Upload file as soon as App is registered***

- (void)onRegisterAppInterfaceResponse:(FMCRegisterAppInterfaceResponse \*)response

{

[…]

if (self.isGraphicsSupported) {

// 获取图像数据

UIImage\* pngImage = [UIImage imageNamed: @"icon.png"]; //Load and Image

NSData\* pngData = UIImagePNGRepresentation(pngImage); //Save as raw Data

// 上传图像数据

FMCFileType\* fileType = [FMCFileType GRAPHIC\_PNG]; //Save Filetype

FMCPutFile\* putRequest = [[FMCPutFile alloc] init]; //Initiate putFile RPC Request

[putRequest setCorrelationID:[NSNumber numberWithInt:autoIncCorrID++]]; //Set CorrelationID

self.appIconPutFileID = [putRequest correlationID]; //Save correlationID to identify completed upload

[putRequest setFileType:fileType];

[putRequest setPersistentFile:[NSNumber numberWithBool:NO]]; //Can be disregarded

[putRequest setSyncFileName:@"icon.png"]; //Handle that will be used to call the image

[putRequest setSystemFile:[NSNumber numberWithBool:NO]]; //Disregarded

[putRequest setOffset:[NSNumber numberWithInt:0]]; //Disregarded

[putRequest setLength:[NSNumber numberWithInt:0]]; //Disregarded

[putRequest setBulkData:pngData]; //Asign the raw data to the rpc request

[proxy sendRPCRequest:putRequest]; //Send out request

}

}

***Select file as soon as upload has finished successfully***

- (void)onPutFileResponse:(FMCPutFileResponse \*)response

{

// 如果response中的correlationID跟appIconPutFileID一致 并且上传成功 则设置图标

if (self.isGraphicsSupported && response.correlationID == self.appIconPutFileID && [response.info isEqualToString: @"SUCCESS"]) {

FMCSetAppIcon\* setAppIconRequest = [[FMCSetAppIcon alloc] init];

[setAppIconRequest setSyncFileName:@"icon.png"]; //Specify Handle

[setAppIconRequest setCorrelationID:[NSNumber numberWithInt:autoIncCorrID++]];

[proxy sendRPCRequest:setAppIconRequest]; //Send request

}

}

***Header***:

@property (strong, nonatomic) NSNumber\* appIconPutFileID; // Keeps track of the correlation ID for putfile

Android code:

@Override

public void onOnHMIStatus**(**OnHMIStatus notification**)**

**{**

**if** **(**mFirstHmiNone **==** **true)** //This variable is set to true in OnProxyOpen

**{**

mFirstHmiNone **=** **false;//该参数需要在新建SyncProxyALM时置为true**

***[SEND FILE AS IN ABOVE EXAMPLE]***

**}**

**[……]**

**}**

@Override

public void onPutFileResponse**(**PutFileResponse response**)**

**{**

//收到PutFileResponse后，我们先检查是否跟上传应用图标的PutFile时的correlationID一致，如果一致 并且上传成功，则调用SetAppIcon设置图标

**if** **(**response**.**getCorrelationID**()** **==** self.appIconCorrID **&&** response**.**getSuccess**())**

**{**

SetAppIcon msg **=** **new** SetAppIcon**();**

//syncFileName就是在PutFile中所设置的syncFileName

msg**.**setSyncFileName**(**“AppIcon.png”**);**

msg**.**setCorrelationID**(**autoIncCorrId**++);**

**try**

**{**

proxy**.**sendRPCRequest**(**msg**);**

**}**

**catch** **(**SyncException e**)** **{**

e**.**printStackTrace**();**

Log**.**e**(**TAG**,** "Exception encountered when trying to send RPC request to SYNC: " **+** e**.**toString**());**

**}**

**}**

**}**

### 上传专辑图片

上传图片跟之前上传图标到车机一样。更新专辑图片的一般流程是：

1. 清除当前显示的图片

在显示专辑图片是需要将之前的图片先清除，以防止在上传图片期间 车机上依然显示之前音乐的专辑图片 ，参考代码如下：

**private** **void** resetGraphiconSync() {

Image image = **new** Image();

image.setValue("");//image中设置syncFileName为空 即可清除当前显示的

image.setImageType(ImageType.*DYNAMIC*);

**try** {

*mSyncProxy*.show(**null**, **null**, **null**, **null**, **null**, **null**, **null**, graphic,

**null**, **null**, **null**, correlationID++);

} **catch** (SyncException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

1. 上传要显示的专辑图片

使用PutFile上传图片，此部分与上传AppIcon有相似之处，不同的是：需要保存一些参数方便 在PutFileResponse中进行判断 然后显示专辑图片，Code参考显示专辑图片的Code。

1. 显示专辑图片

在PutfileResponse中 通过判断 correlationID以及response.success 来确认上传专辑图片已经成功，之后调用Show显示专辑图片 参考代码如下：

**在收到用户切换歌曲的指令后 应马上调用PutFile上传专辑图片，在上传后 使用变量标记 需要显示专辑 和 上传时的CorrelationID，方便在收到PutFileResponse 时调用Show显示专辑图片**

//[…] 在调用PutFile后 保存以下PutFile相关参数

self.coverArtHandle = @"coverArt.png";//PutFile中syncFileName

self.coverArtToDisplay = TRUE;

self.coverArtCorID = [putRequest correlationID]; //PutFile中correlationID

- (void)onPutFileResponse:(FMCPutFileResponse \*)response

{

[…]

//上传专辑图片成功并且需要显示专辑图片时 利用之前保存的参数显示图片

if (self.coverArtToDisplay == TRUE &&

[ response.correlationID isEuqalToNumber:self.coverArtCorID &&

[response.resultCode isEqual: [FMCResult SUCCESS]) {

FMCShow\* newShow = [[FMCShow alloc] init];

FMCImage\* newImage = [[FMCImage alloc] init];

[newImage setValue: self.coverArtHandle];

[newImage setImageType:[FMCImageType DYNAMIC]];

[newShow setGraphic;newImage];

[newShow setCorrelationID: [NSNumber numberWithInt: autoCorrelationID++]];

[proxy sendRPCRequest:newShow];

self.coverArtCorID = FALSE;

}

}

***Header:***

@property (strong, nonatomic) NSNumber\* coverArtCorID; // 保存PutFile的CorrelationID

@property (assign) BOOL coverArtToDisplay; // 是否需要在PutFile成功后显示专辑图片

@property (assign) NSString\* coverArtHandle;//PutFile 上传图片到SYNC 图片的名称

**在收到用户切换歌曲的指令后 应马上调用PutFile上传专辑图片，在上传后 使用变量标记 需要显示专辑 和 上传时的CorrelationID，方便在收到PutFileResponse 时调用Show显示专辑图片**

// […]在调用PutFile后 保存以下PutFile相关参数

coverArtToDisplay **=** **true;**

coverArtCorID **=** putRequest**.**getCorrelationID**();**//PutFile中correlationID coverArtHandle **=** putRequest**.**getSyncFileName**();**//PutFile中syncFileName

@Override

public void onPutFileResponse**(**PutFileResponse response**)** **{**

// […] Other Code

//上传专辑图片成功并且需要显示专辑图片时 利用之前保存的参数显示图片**if** **(**coverArtToDisplay **&&** response**.**getCorrelationID**().**equals**(**coverArtCorID**)** **&&** response**.**getSuccess**())** **{**

Show newShow **=** **new** Show**();**

Image newImage **=** **new** Image**();**

newImage**.**setValue**(**coverArtHandle**);**

newImage**.**setImageType**(**ImageType**.**DYNAMIC**);**

newShow.setGraphic(newImage);

newShow.setCorrelationID(autocorrelationID++);

// Remember to add a try/catch block

proxy**.**sendRPCRequest**(**newShow**);**

coverArtToDisplay **=** **false;**

**}**

**}**

***Header:***

private boolean coverArtToDisplay**;**

private int coverArtCorID**;**

private String coverArtHandle**;**

### 设置进度条

Gen3上支持显示进度条，一般使用步骤如下，应用收到音乐切换提示后 ：

1. 清除进度条
2. 设置进度条
3. 如果中间暂停可以暂停进度条，或者 继续进度条

如果当前没有音乐播放应该清楚进度条，具体使用方法如下：

**//设置进度条**

SetMediaClockTimer mTimer = **new** SetMediaClockTimer();

StartTime starttime = **new** StartTime();

starttime.setMinutes(minutes);

starttime.setSeconds(seconds);

starttime.setHours(0);

StartTime endtime = **new** StartTime();

endtime.setMinutes(length / 60);

endtime.setSeconds(length % 60);

endtime.setHours(0);

mTimer.setStartTime(starttime);

mTimer.setEndTime(endtime);

mTimer.setCorrelationID(correlationID++);

mTimer.setUpdateMode(UpdateMode.*COUNTUP*);

*mSyncProxy*.sendRPCRequest(mTimer);

//暂停进度条

SetMediaClockTimer mTimer = **new** SetMediaClockTimer();

StartTime starttime = **new** StartTime();

starttime.setMinutes(0);

starttime.setSeconds(0);

starttime.setHours(0);

mTimer.setStartTime(starttime);

mTimer.setCorrelationID(correlationID++);

mTimer.setUpdateMode(UpdateMode.*PAUSE*);

*mSyncProxy*.sendRPCRequest(mTimer);

//清除进度条

*mSyncProxy*.setMediaClockTimer(**null**, **null**, **null**, UpdateMode.*CLEAR*,

correlationID++);

//继续进度条

*mSyncProxy*.setMediaClockTimer(**null**, **null**, **null**, UpdateMode.*RESUME*,

correlationID++);

**//设置进度条**

**FMCSetMediaClockTimer \*scmt = [[FMCSetMediaClockTimer alloc] init];**

**scmt.startTime = [self \_timeIntervalToStartTime:time];**

**scmt.endTime = [self \_timeIntervalToStartTime:[\_streamer duration]];**

**scmt.updateMode = [FMCUpdateMode COUNTUP];**

**scmt.correlationID = [NSNumber numberWithInt:autoIncCorrID++];**

**[proxy sendRPCRequest:scmt];**

**- (FMCStartTime\*)\_timeIntervalToStartTime:(NSTimeInterval) time{**

**int hour = time/3600;**

**int minutes = (time - hour\*3600)/60;**

**int seconds = time-hour\*3600-minutes\*60;**

**FMCStartTime\* starttime = [[FMCStartTime alloc] init];**

**starttime.hours = [NSNumber numberWithInt:hour];**

**starttime.minutes = [NSNumber numberWithInt:minutes];**

**starttime.seconds = [NSNumber numberWithInt:seconds];**

**return starttime;**

**}**

//暂停进度条

[proxy sendRPCRequest:[FMCRPCRequestFactory buildSetMediaClockTimerWithUpdateMode:[FMCUpdateMode PAUSE] correlationID:[NSNumber numberWithInt:autoIncCorrID++]]];

//清除进度条

[proxy sendRPCRequest:[FMCRPCRequestFactory buildSetMediaClockTimerWithUpdateMode:[FMCUpdateMode CLEAR] correlationID:[NSNumber numberWithInt:autoIncCorrID++]]];

//继续进度条

[proxy sendRPCRequest:[FMCRPCRequestFactory buildSetMediaClockTimerWithUpdateMode:[FMCUpdateMode RESUME] correlationID:[NSNumber numberWithInt:autoIncCorrID++]]];

### 在SoftButton和AddCommand PerformInteraction中使用图片

SoftButton AddCommand PerformInteraction中都支持显示图片，具体方法跟以上基本一致，首先上传合适的图片到SYNC，然后 设置SoftButton AddCommand 或Choice中的Image参数，发送Show、AddCommand、CreateInteractionChoiceSet 即可。

### 设置帮助列表

设置帮助列表可以使用SetGlobalProperties，示例代码如下：

Android代码:

NSString \*vrHelpString = @"";

//初始化一个数组，用于存储所有的语音命令

NSMutableArray \*vrHelpArray = [[NSMutableArray alloc]init];

for (int i = 0; i < [commandVrArray count]; i ++) {

//初始化

FMCVrHelpItem \*helpItem = [[FMCVrHelpItem alloc]init];

//图片nil

helpItem.image = nil;

//设置语音提示文本内容（commandVrArray语音命令集合）

helpItem.text = [commandVrArray objectAtIndex:i];

//位置

helpItem.position = [NSNumber numberWithInt:i+1];

//添加到数组中

[vrHelpArray addObject:helpItem];

//拼接语音命令

vrHelpString = [vrHelpString stringByAppendingFormat:@"%@,",[commandVrArray objectAtIndex:i]];

}

FMCSetGlobalProperties \*globalProperties = [FMCRPCRequestFactory

buildSetGlobalPropertiesWithHelpText:vrHelpString//帮助提醒，所有语音命令逗号分隔

timeoutText:vrHelpString//超时提醒

vrHelpTitle:@"网络音乐帮助"//对话框标题

vrHelp:vrHelpArray//所有语音帮助

correlationID:[NSNumber numberWithInteger:\_autoIncCorrID++]];

//send

[\_syncProxy sendRPCRequest:globalProperties];

**Vector<VrHelpItem> items = new Vector<VrHelpItem>();**

**StringBuilder builder = new StringBuilder();**

**int count = COMMANDS.length;**

**for (int j = 0; j < count; j++) {**

**VrHelpItem item = new VrHelpItem();**

**item.setPosition(j+1);**

**item.setText(getString(COMMANDS[j]));**

**item.setImage(null);**

**builder.append(getString(COMMANDS[j]+ ",");**

**items.add(item);**

**}**

**try {**

**mSdlProxy.setGlobalProperties(**

**TTSChunkFactory.**

**createSimpleTTSChunks(builder.toString()),//帮助提醒，所有语音命令逗号分隔**

**TTSChunkFactory.**

**createSimpleTTSChunks(builder.toString()),//超时提醒**

**"网络音乐帮助", //对话框标题**

**items,//所有command的第一个语音**

**correlationID++);**

**} catch (SdlException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

## SYNC Gen3 Layout介绍

### 默认媒体类应用屏幕Layout



没有进度条



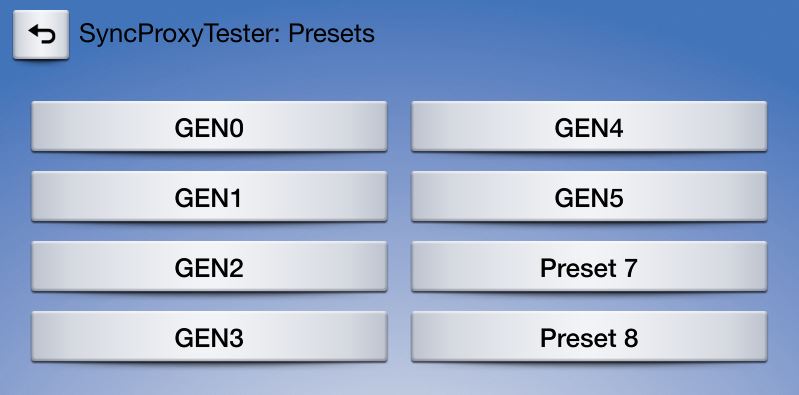
图片大小为:

高度: 185 px

宽度: 185 px

SoftButton 上支持的图片大小为： 35px 35px

应用菜单：预设按钮界面 :



### 其他可用Layout

以下Layout可以通过proxy.setDisplayLayout RPC来设置，通过此方法可以根据自身特点选择不同的界面，例如 天气类应用可以选择??????

android code example:

proxy.setdisplaylayout("NON-MEDIA", autoIncCorrId++);

NON-MEDIA



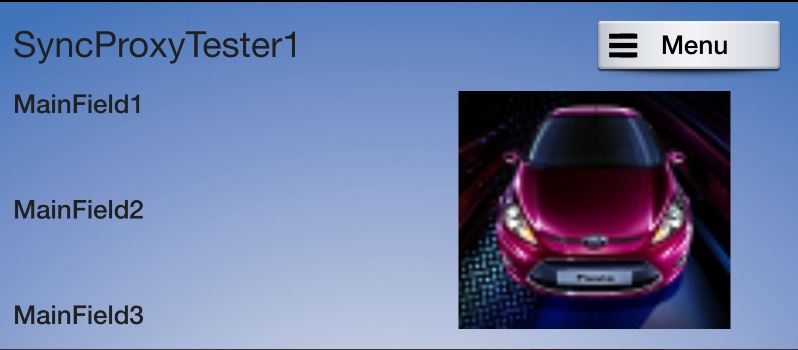
NON-MEDIA (With SoftButtons)



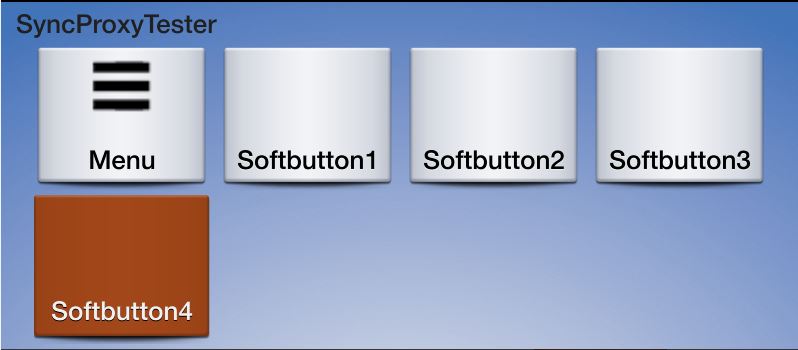
GRAPHIC\_WITH\_TEXT



TEXT\_WITH\_GRAPHIC



TILES\_ONLY



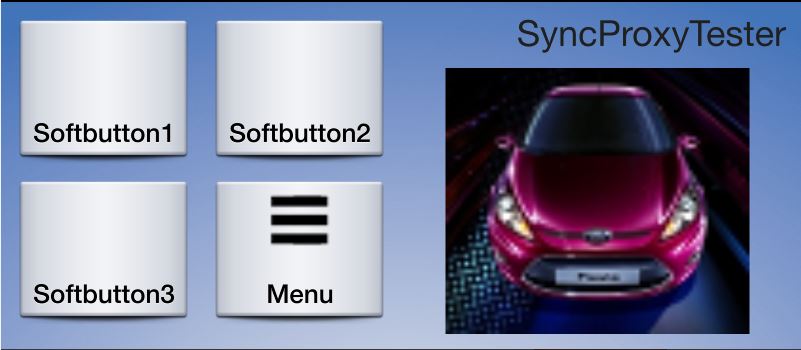
The softbuttons on this template are 70px by 70px

GRAPHIC\_WITH\_TILES



The softbuttons on this template are 70px by 70px

TILES\_WITH\_GRAPHIC



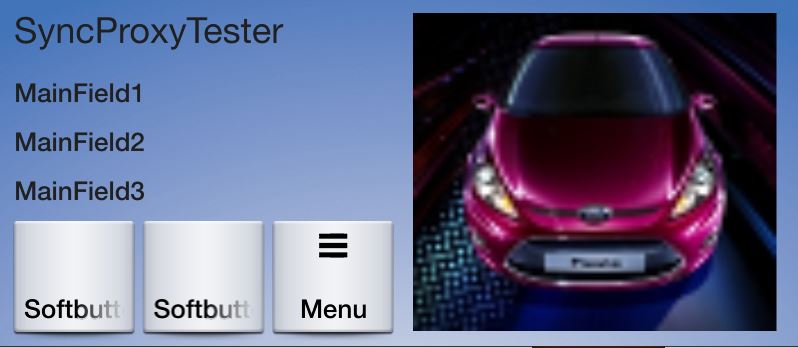
The softbuttons on this template are 70px by 70px

GRAPHIC\_WITH TEXT\_AND\_SOFTBUTTONS



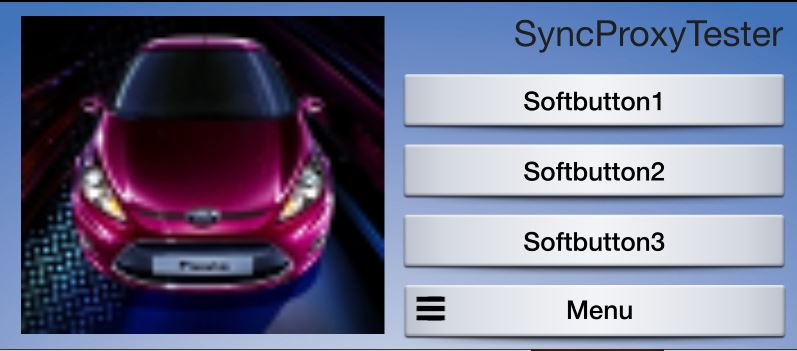
The softbuttons on this template are 70px by 70px

TEXT\_AND\_SOFTBUTTONS\_WITH\_GRAPHIC



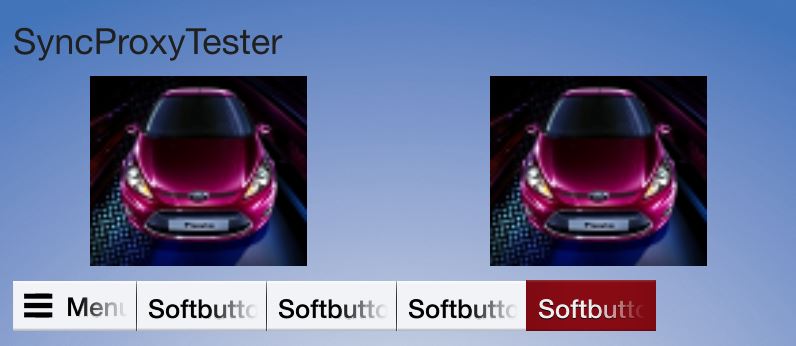
The softbuttons on this template are 70px by 70px

GRAPHIC\_WITH\_TEXTBUTTONS



The softbuttons on this template are 35px by 35px

DOUBLE\_GRAPHIC\_SOFTBUTTONS



The following image specifications are required for “DOUBLE\_GRAPHIC\_WITH\_SOFTBUTTONS”

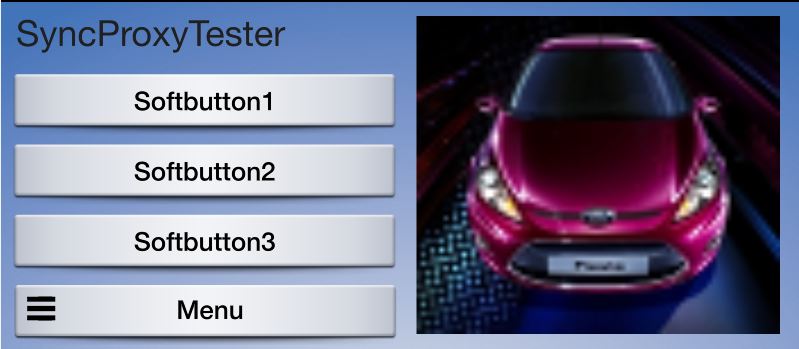
half screen with softkeys

"resolutionHeight": 250,

"resolutionWidth": 370

The softbuttons on this template are 35px by 35px

TEXTBUTTONS\_WITH\_GRAPHIC



The following image specifications are required for “TEXT\_BUTTONS\_WITH\_GRAPHIC”

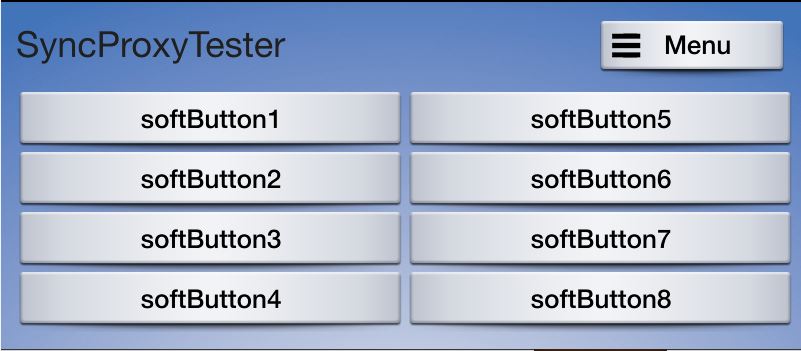
Half screen with appName and softkeys

"resolutionHeight": 210,

"resolutionWidth": 370

The softbuttons on this template are 35px by 35px

TEXTBUTTONS\_ONLY



The softbuttons on this template are 35px by 35px

**LARGE\_GRAPHIC\_WITH\_SOFTBUTTONS**

The following image specifications are required for “LARGE\_GRAPHIC\_WITH\_SOFTBUTTONS”

Full width, with softkeys

"resolutionHeight": 210,

"resolutionWidth": 770

The softbuttons on this template are 35px by 35px

**LARGE\_GRAPHIC\_ONLY**



RPC graphical representations

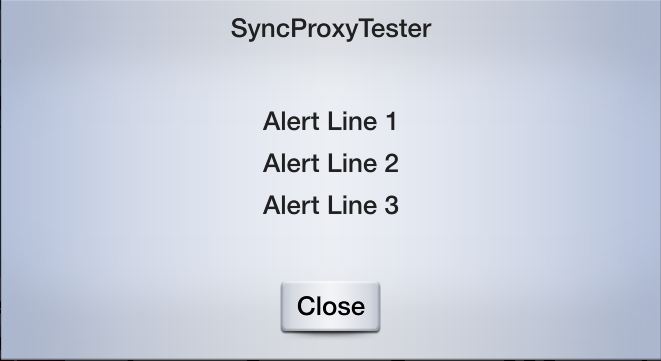
Alert()

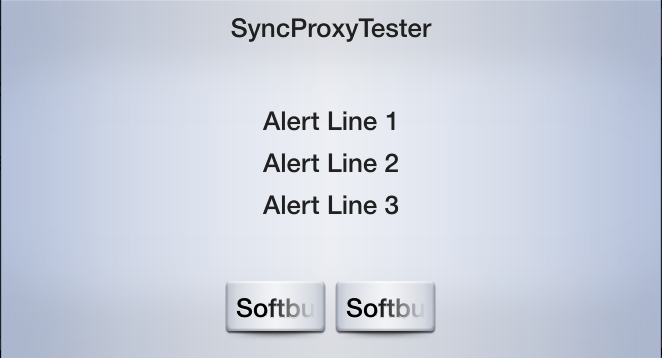
Shown with no or 1 softbutton – up to 4 softbuttons:

Note that while the max is 3 lines of text, not all lines must be populated.

### API Layout 介绍

#### ALERT WITH NO SOFTBUTTONS





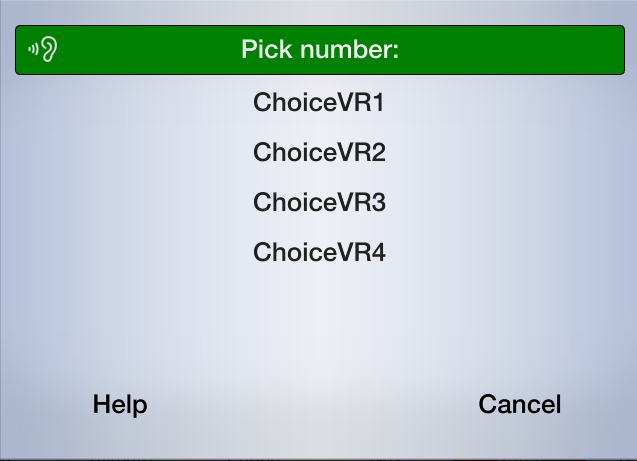




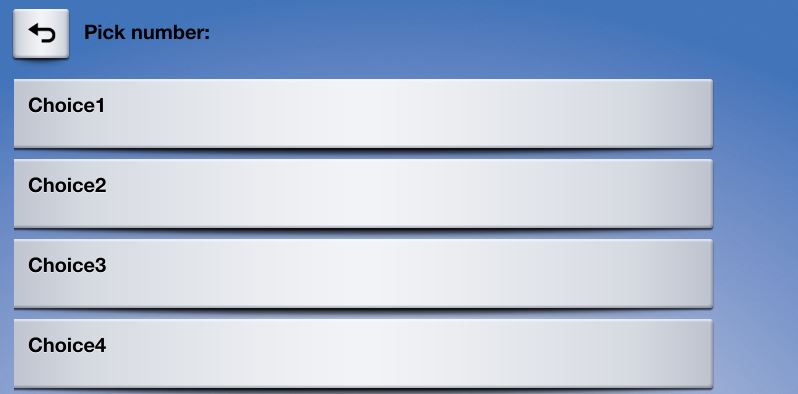
#### PerformInteraction

Note that the InteractionMode.BOTH mode starts with the voice session and if that is canceled or the screen is touched, it reverts to the MANUAL\_ONLY screen.

InteractionMode.VR\_ONLY

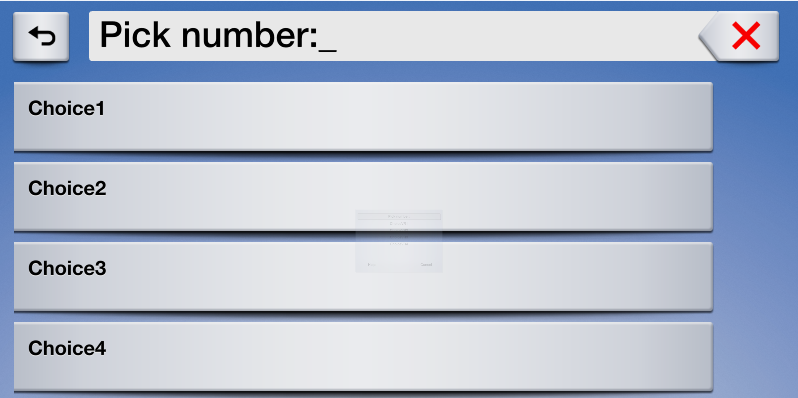


InteractionMode.MANUAL\_ONLY

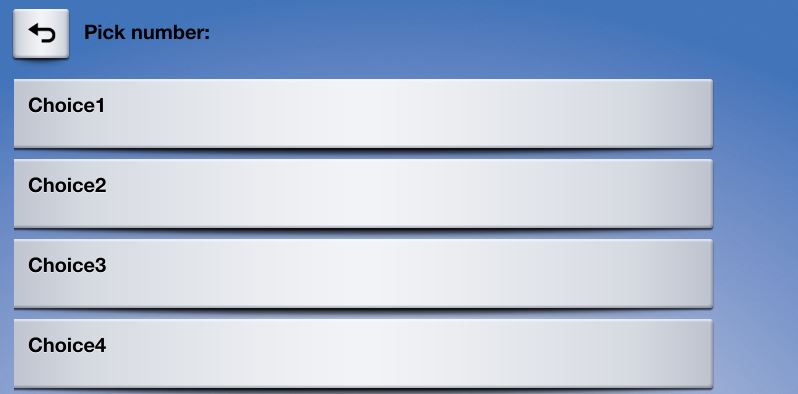


Additional perform interaction layouts:

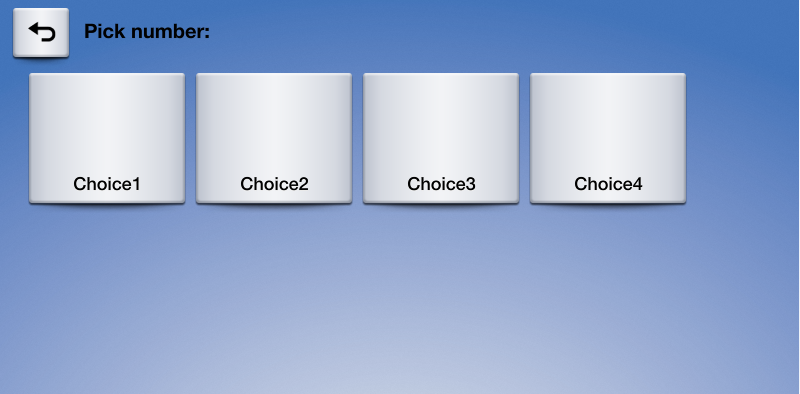
LIST\_WITH\_SEARCH



LIST\_ONLY



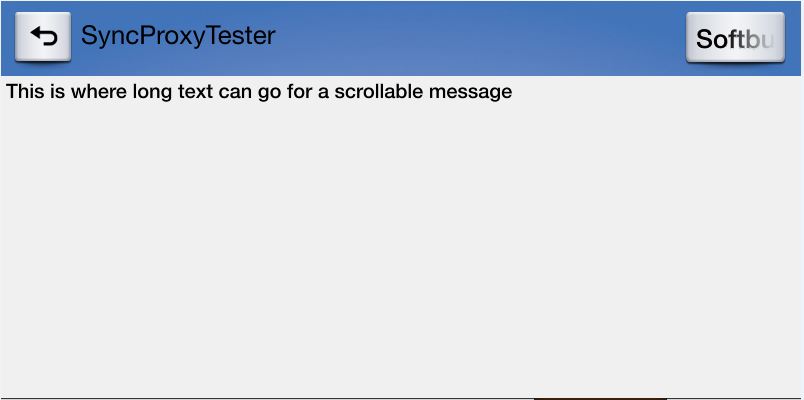
ICON\_ONLY



The softbuttons on this templates are 70px by 70px

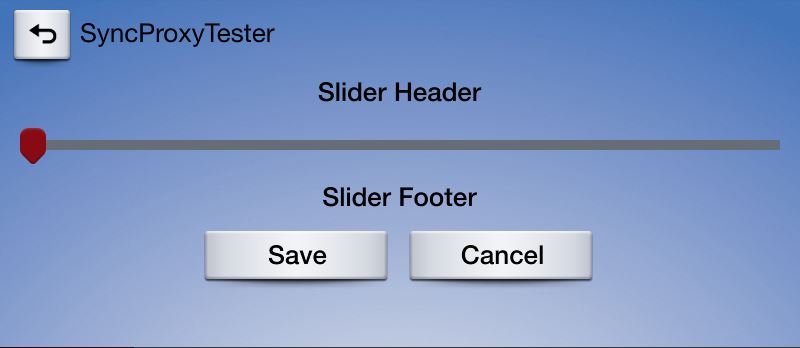
#### ScrollableMessage

\*Note that this is only available while the vehicle is not in motion.

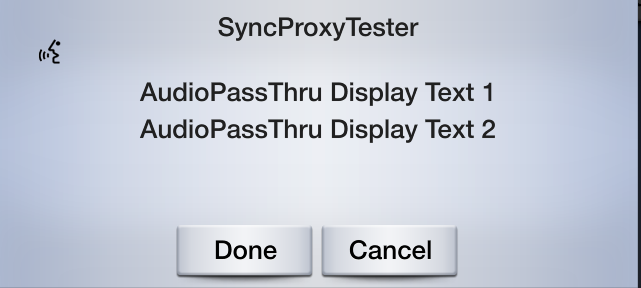


#### Slider

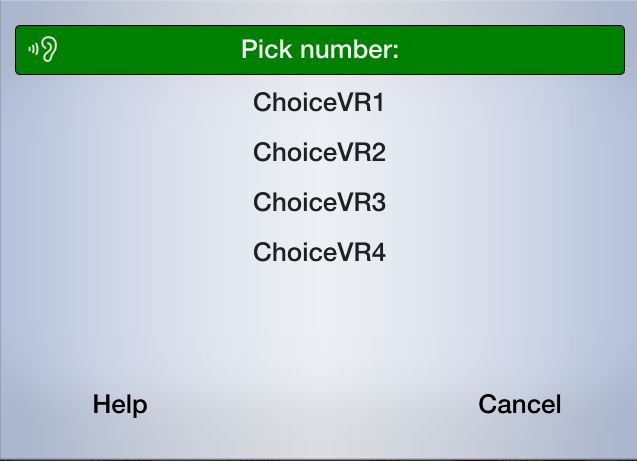
\*Note that this is only available while the vehicle is not in motion.



#### AudioPassThru



#### Voice Help Pop-up



## 图片规格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ImageName** | **Used in RPC** | **Details** | **Height** | **Width** | **Type** |
| softButtonImage | show | Will be displayed on softbuttons on the basescreen | 35px  or  70px | 35px  or  70px | png,jpg,bmp |
| choiceImage | createInteractionChoiceSet | Will be displayed in the manual part of an performInteraction either big (ICON\_ONLY) or small (LIST\_ONLY) | 35px  or  70px | 35px  or  70px | png,jpg,bmp |
| choiceSecondaryImage | createInteractionChoiceSet | Will be displayed on the right side of an entry in (LIST\_ONLY) performInteraction | 35px | 35px | png,jpg,bmp |
| vrHelpItem | setGlobalProperties | Will be shown during voice interaction | 35px | 35px | png,jpg,bmp |
| menuIcon | setGlobalProperties | This is shown on the “More…” button | 35px | 35px | png,jpg,bmp |
| cmdIcon | addCommand | Will be shown for commands in the "More…" menu | 35px | 35px | png,jpg,bmp |
| appIcon | setAppIcon | Will be shown as Icon in the "Mobile Apps" menu | 70px | 70px | png,jpg,bmp |
| Graphic (size will be different in different layout) | show | Will be shown on the basescreen as coverart | 185px | 185px | png,jpg,bmp |