General Motors

Infotainment SDK Version 2.1.0_01

Release Notes

Release Notes

1. Introduction

1.1 Copyright Notice

Copyright © GM
Copyright 2012 GM Global Technology Operations LLC.
All rights reserved.

1.2 Naming Convention

The following is the naming convention for the GM SDK.

SDK v**X.X.X_YY**

- The first 3 digits (X.X.X) denote the major version and will be based on an internal GM API spec.
- The last 2 digits (YY) denote the minor version and will start from 01 for the first release of the SDK within that API specification

1.3 GM Infotainment SDK

GM Infotainment SDK (also referred to in this document as "GM SDK") is intended to provide support for Application Development for supported GM in-vehicle devices. The GM SDK includes the following features:

1. Rich set of APIs.
   a. Developers can use this API set to retrieve vehicle information and create smart applications for the GM in-vehicle devices. Refer to SDK documentation for more details.
2. Widgets
   a. Developers can use the APIs provided to create standard widgets like "keyboard", "number pad", "date picket", etc. Refer to the SDK documentation for more details.
3. Code Snippets
   a. Code snippets that developers can reuse during application development.
4. Help Contents
   a. Detailed API help documentation
5. Emulator
   a. Developers can test their applications using the GM SDK Emulator that is included in the GM SDK.

> NOTE 1: The applications developed using this release of the GM SDK are intended to be installed and run on supported GM in-vehicle devices only.

> NOTE 2: Eclipse IDE is required to install and develop applications using GM SDK.

1.4 Purpose

The purpose of the Release Notes is to communicate the new features and changes in this release of the Infotainment Developers Toolkit (SDK v2.1.0_01). It also documents known problems and workarounds.

1.5 Scope

This document describes the SDK v2.1.0_01 release.

2. Release Content

2.1 Release Summary

SDK Release	Date Released
V2.1	12/12/2012

Widgets

Widgets are added to GM's existing Widgets library. Developers can make use of these Widgets for the "standard ready to use functionality".

The following widgets are rewritten to be more lightweight on the radio:

Button, Checkbox, RadioButton, Textfield, Textspinner, CloseApp, Popup, SetFavorite, AboutApp, Keyboard, and Numberpad.

IMPORTANT NOTE:

Developers using an older version of GM SDK need to update the way they instantiate the widgets in their existing apps to reflect the new lightweight widgets implementation.

For more information regarding the Keyboard and the numberpad widgets, please refer to one of the following resources:

1. Code Snippets in Eclipse IDE
2. Help Content: (Eclipse -> Help -> Help Content -> GM Developer's Library)
3. Code Sample Project: (Eclipse -> File -> New -> New Dashboard Web Application Project)

Emulator
  ➢ Redesigned the Rotary knob to be more efficient and easy to use. The current rotary knob implementation generates one event for each knob move.

Defect Fixes – The following defects (identified in the previous release) are fixed.

  ➢ **PLS URLs Support in Emulator:** Emulator now supports PLS URLs which matches radio systems support list.

  ➢ **Code Sample Project:** Fixed a bug in Code Sample App related to the Keyboard Widget

  ➢ **Connection Status:** The default connection status in the emulator settings is set to Active (3) instead of Not Connected (0).

  ➢ **API Fixes:** Fixed different APIs and their corresponding Code Snippets.

API Changes – *Added/Removed/Updated API methods*
Click on the Icon below to view the supported APIs

https://developer.gm.com/sites/default/files/vehicle-files/vehicle-zip/Documentation/GM_SDK_API_Changes.pdf

Note – *All Apps have to invoke gm.system.init during onload event (start-up) of the App. For the new projects that are going to be created using the GM SDK Plugin, it will be automatically added. However, for the existing projects (if it is not already done) this method call has to be added in the "onload" event. Please be aware that failing to invoke this method during startup will result in the failure of some of the API Methods.*

Code Snippets –
  ➢ *Code Snippets corresponding to the new/missing API methods and the new widgets are added.*
  ➢ *Enhanced Code Snippets corresponding to the new Keyboard and Numberpad Widgets are added.*
  ➢ *Issues with existing Code Snippets are resolved.*

Help –
  ➢ *Help corresponding to the new/missing API methods are added*
  ➢ *Issues with existing Help are resolved*

Others -
  ➢ *License Agreement, Feature Description and Copyright Information were removed from the SDK*
  ➢ *Yelp App is removed from the GM SDK*

3. Known Issues/Limitations

The table below lists currently known issues and any limitations for this version of SDK that developers will have to work around till a future release.

Issues	Details
1	Certain API methods are not supported in the emulator. Please refer to the spreadsheet (GM_SDK_API_Changed) that is attached under section 2.1 – API Changes
2	The following widgets do not support rotary events for Non-Touch screens. They are design especially for Touch-Screens and based on touch gestures: Toggle Button / List Selector / Scrolling Selector / Page Flicker
3	Not all Enum values are available in GM Constants
4	GetAID, getWayPoint, getMediaPlayerList, getMediaPlayerResultNum, seek, setInteractionSelector methods do not work. Those APIs will be implemented in future releases
5	Favorite Links in the Emulator does not work
6	Not all locales are supported and we are working on creating a list of supported locales
7	getResource API searches only in the 2 following locations: **Data, Local.** If the [number] exceeds the number of files available, then the first file will be returned.
8	Keyboard widget supports only the following languages: English: "en-US" | French: "fr-CA" | Arabic: "ar-WW". For more information check the Keyboard section in this document

4. Installation

The GM SDK software is shipped as a zip file and the zip file can be installed/updated into Eclipse IDE as-is. Please refer to Eclipse documentation for more details on how to install/update software in Eclipse IDE.

NOTE: If you have a previous version of GM SDK installed, you need to uninstall it before trying to install this current version of the SDK.

For more information, please refer to the "GM SDK Installation Guide".

https://developer.gm.com/sites/default/files/vehicle-files/vehicle-zip/Documentation/GM_Infotainment_SDK_Setup_Guide.pdf

5. **Issues/Bugs/Access**

Any issues, concerns or questions regarding the GM SDK should be posted to the GM SDK developer forum

6. Frequently Asked Questions:

1. Audio playback is not working, what is the problem?
There are some problems running Applets in Chrome browser on Mac after your latest software update. This is happening due to incompatibilities between JDK 7 64-bit on Chrome.
You need to downgrade your JDK. Please follow the instructions on how to downgrade your JDK to Apple JDK. http://support.apple.com/kb/HT5559

2. How I can initialize the keyboard widget?

To create a keyboard widget you need to follow the following steps:
      1. Create a div in your HTML page that has the input tag.
      Example:

```
<div id="keyboardDiv">
      <label class="inputLabel"> Placeholder </label>
      <input type="text"></input>
</div>
```

      NOTE: The Label tag has a class named "inputLabel" to act as a placeholder for the input field when you load the keyboard widget

      2. Initialize the keyboard in your JavaScript file.
      Example:

```
var keyboard = new gm.widgets.Keyboard({sender: "divID", language: "en-US",
kbType: 1, feedbackMode: 0});
```

3. How I can initialize the keyboard widget?

1. Create a div in your HTML page that has the input tag.
Example:

```
<div id="numberpadDiv">
      <label class="inputLabel"> Placeholder </label>
      <input type="text"></input>
</div>
```

2. Initialize the Numberpad in your JavaScript file.
Example:

```
var numberpad = new gm.widgets.Numberpad({sender: "divID", language: "en-
US"});
```

Common Mistakes with the Keyboard and Numberpad Widgets:

Wrong Way

```
<div id="divID">
      <form name="formName" action="formAction">
            <label class="inputLabel"> Placeholder </label>
            <input type="text"></input>
      </form>
</div>
```

Right Way

```
<form name="formName" action="formAction">
      <div id="divID">
            <label class="inputLabel">Placeholder</label>
            <input type="text"></input>
      </div>
</form>
```

For more information regarding the Keyboard and the numberpad widgets, please refer to one of the following resources:

4. Code Snippets in Eclipse IDE
5. Help Content: (Eclipse -> Help -> Help Content -> GM Developer's Library)
6. Code Sample Project: (Eclipse -> File -> New -> New Dashboard Web Application Project)

4. How to handle the Rotary Knob Events for non-touch screen?

Currently, the system is handling those events for all the apps. But developers can handle it by using the following API **watchRotaryControl (success , [failure]).** It is advised to call **clearRotaryControl (rotaryHandle)** after using **watchRotaryControl**.

If you prefer the system to handle those events for your app, you need to make sure that elements in the body tag that require Rotary Events (for non-touch screens) are one of the following:

INPUT, **TEXTAREA**, **BUTTON**, or any other HTML tag with **onClick**, **onMouseDown**, or **onMouseUp**.

For more information please refer to (Eclipse -> Help -> Help Content -> GM Developer's Library) for more details.