

Technical Guidelines For GM Infotainment App Developers

Version 1.2



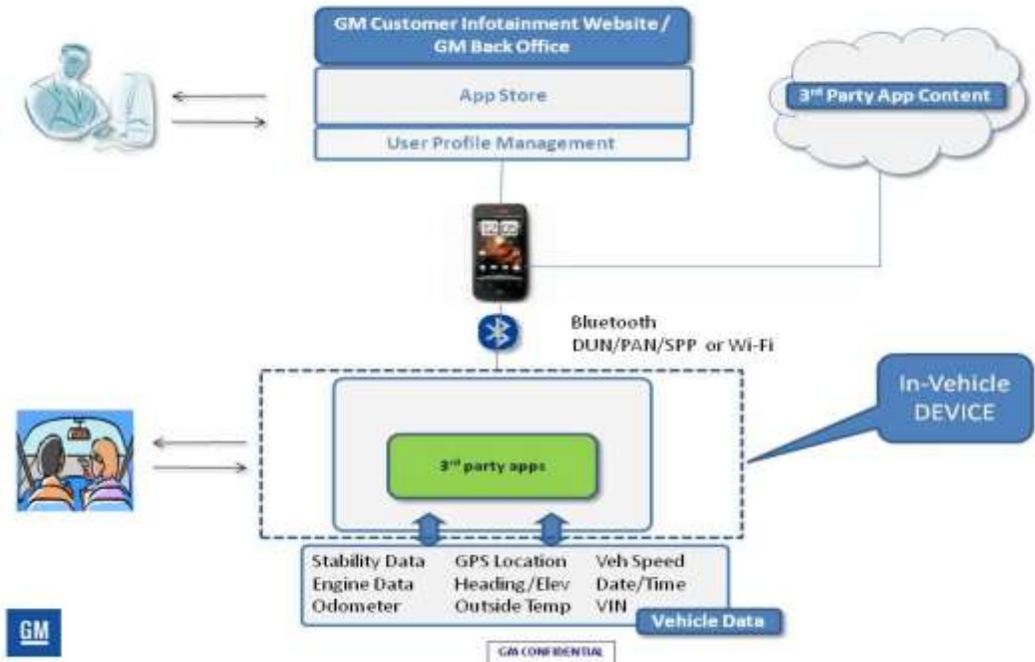
Contents

1. GM Infotainment Apps Overview	4
2. Framework Overview.....	4
3. Overview of App development cycle for the third party developer	5
4. Infotainment Platforms	6
5. Customer Data and Privacy.....	Error! Bookmark not defined.
6. App Requirements	10
6.1. General	10
6.2. App Packaging Requirements.....	11
6.3. Functional Requirements	12
6.3.1. About Screen	12
6.3.2. Other.....	12
6.4. App Data Folder Requirements.....	13
7. App Submission Requirements.....	13
8. Coding requirements	13
9. Communication Protocol.....	14
10. Security.....	14
11. Audio/Video Playback	15
12. App Resource Utilization.....	16
13. App Performance	16
14. App fonts	16
15. App Design, System Interaction and safety and validation requirements.....	16
16. App Usability	16
17. Example App Code Structure.....	17
18. App Configuration	17

19. App Configuration Possible Values	20
20. GM SDK.....	20
21. GM SDK - Demo App / Sample Code.....	21
22. Living Document.....	21

1. GM Infotainment Apps Overview

GM's infotainment system provides connected-car functions by linking the motorists' smart phone to the vehicle's in-dash infotainment system. Customers will be able to choose GM certified Apps that comply with GM's HMI (Human Machine Interface) standards for safe driving experience. A browser based framework is developed by GM to enable customers to download GM certified apps to their vehicle. Internet connectivity to the vehicle infotainment system can be established via customer's smart phone using short range wireless communication (e.g. **Bluetooth**® DUN/PAN/SPP or Wi-Fi). GM allows customers to download Apps into their USB from the GM back office and upload it into the vehicle infotainment system. Once a phone connection is established with the product and the user has registered into the GM back office, the user will be able to download GM certified apps from GM's back office or AppShop (for Buick, Chevrolet and GMC; different name for Cadillac) via the data pipe. The diagram below illustrates the infotainment system.



Note: Multiple connection methods are transparent to the App developers.

2. Framework Overview

The Internet Apps Framework is a browser framework that manages and controls how internet Apps are downloaded to, installed on, and run on an infotainment system in the

vehicle. The Internet Apps are displayed and formatted utilizing an HTML rendering engine and JavaScript engine. The Internet Apps Framework manages and controls all instances of HTML rendering windows and displays the desired URL in the appropriate rendering window. Apps require local resource storage. The Framework manages all local resources associated with Internet Apps and implements an interface for launch and control of the Internet Apps. The Internet Apps Framework interfaces with other functionality so that Apps are able to perform many functions such as text-to-speech operations available on the infotainment system, Speech Recognition(in works), source audio, play video, store favorites, store destinations and POIs, and communicate with a connected phone.

There are two main types of functionality exposed: vehicle data and infotainment system interaction. The Browser Framework has the capability of providing the list of Apps upon request and synchronizes with the GM AppShop (for Buick, Chevrolet and GMC; different name for Cadillac). This synchronization process shall provide a mechanism to download new Apps, update existing Apps, and delete Apps. The App transfer process shall be initiated by the Browser/Internet Apps Framework App Manager. Using HTTPS requests and the necessary credentials will need to be provided for any transfer to take place.

3. Overview of App development cycle for the third party developer

To develop an App for GM's infotainment platform, developers must first execute the Developer Agreement (Cover Letter Agreement) for the GM APP SDK. After the developer has agreed to the Developer Agreement for the GM APP SDK, GM will provide the details for the developer to download the latest version of the GM SDK and supporting documentation. Apps can be built for the infotainment platform using the APIs available in the GM SDK. Since GM's Infotainment platform is browser based, Apps must be built primarily using HTML5 ("GM_Infotainment_HTML5_Support.pdf" lists the HTML5 tags supported) & JavaScript. The emulator provided in the SDK must be used to test the App. The App must be exported from the GM SDK and submitted to GM along with the details about the App.

Upon submission of the App with all the requisite details, GM tests the App to verify that the App conforms to certain technical requirements. Once the App is certified to be approved, then the App is available to be posted in GM's App-store. Any issues during certification process will be communicated to the App developer.

Apps may be resubmitted after issues have been addressed.

4. Infotainment Platforms

The different platforms available for Infotainment Systems can be differentiated using the API levels. API v1.0 & API v1.5 are available for MY2013 vehicles. MY2014 vehicles will all have API v2.x.

The Apps can use the GM API function `getVehicleConfiguration` to obtain the build configuration and features available on the vehicle. Apps can use this information to customize their functionality. Some advanced touch gestures like swipe, pinch, fling, spread are not available on some touch screen infotainment systems. Some systems also have rotary knob to scroll and select. In order to support the scrolling select, apps must watch for the event and have tab sequence/orders for the elements on the screen. Apps must have functionality to support all the input methods.

The language that is currently selected can be received from the `getVehicleConfiguration` API. “GM_Infotainment_Language_Support.pdf” lists all the languages that the vehicles might support.

The [GM SDK](#) Section below mentions how to switch between APIs in the SDK.

The differences between API 1.0 and API 1.5 are listed below.

Feature	API v1.0	API v1.5
Screen Size/Resolution/Screen Type	8 inch / 800x480 pixels / Touch Screen 4.2 inch /480x272 /Non Touch Screen	7 inch / 800x480 pixels / Touch & Non Touch Screen
Basic Touch Gestures converted to Mouse	X	X
Advanced Touch Gestures converted to Mouse (swipe, fling, pinch, spread)	X	
Apps as Favorites	X	

USB Install	X	X
Graphics	Canvas 2D Graphics	Canvas 2D Graphics and WebGL
Alerts	Notifications Only	X
Data Folder Size	5MB	5MB
Max App Size (App package)	5MB	5MB
App Icon Friendly Name Length (Characters)	12 characters	TBD
Voice Recognition (In Works)	Off Board	Off Board
Browser	QTWebkit v2.2 based on QT v4.7.4	Open Webkit Project release 92273
Back Button on the Faceplate		X
Back Button in Status area (upper 90 pixels) on Touch Screen		X
Vehicle Data	Listed in "GM_Infotainment_Vehicle_data_details.pdf"	

5. Customer Data and Privacy

5.1. The App must include a privacy statement that describes the type of data collected by the App, how that data will be used, and to whom that data will be disclosed or shared consistent with the Mobile Marketing Association's Privacy Principles Framework. This privacy statement must be presented and accepted by to the user prior to a user being able to utilize the App. The Mobile Marketing Association's Privacy Principles Framework can be downloaded on at: [http://www.mmaglobal.com/MMA_Mobile_App_Privacy_Policy_15Dec2011PC_Update_FINAL\[1\].pdf](http://www.mmaglobal.com/MMA_Mobile_App_Privacy_Policy_15Dec2011PC_Update_FINAL[1].pdf)

5.2. The App must present the user with a way to view the privacy statement at a later time.

- 5.3. Users must provide their consent prior to the App collecting any personal information. Personal information may include, but is not limited to: username, contact information, UDID or VIN. Users must have the right to access, correct and/or delete their personal information, as required by law.
- 5.4. Upon first execution of the App, Users must provide their consent prior to the App collecting any precise geolocation information (including but not limited to the APIs `getCurrentPosition`, `watchPosition`, `getVehicleData` (location data), `watchVehicleData` (location data), `getDestination`, `setDestination`, `getManeuverList`). Users must be allowed the opportunity to revoke that consent at any time.
 - 5.4.1. For users in Greece, Slovakia, Poland, Spain, Korea, Belgium, and Poland, the App must obtain consent from the user for the capture and use of precise geolocation data **per ignition cycle** (upon turning the ignition key) if/when the App is open. A user must be allowed the opportunity to revoke that consent at any time.
 - 5.4.2. For users in Belgium and Poland, the App must include a constant notification or persistent indicator that will alert the user that precise geolocation data is being collected by the App. A user must be allowed the opportunity to revoke that consent at any time.
 - 5.4.3. For users in Korea, geolocation data must be encrypted in transit and at rest.
 - 5.4.4. If the App will collect precise geolocation information in Korea, you are required to first: a) receive approval as Location Service Provider by filing a report with the Korean Communications Commission; and b) obtain a license as a Location Information Provider from the Korean Communications Commission.
- Note: country API will be provided in a later release.*
- 5.5. Appropriate physical, electronic and administrative safeguards to protect the Personal Information should be maintained. Personal information must be sent and secured on a secure channel.
- 5.6. Information collected by the App should not be catalogued by VIN.
- 5.7. If a user revokes their consent, the App must facilitate the user's request.
- 5.8. If a user rejects/revokes the user consent request, the App can either disable the features that use this data or shutdown the App. If necessary, an appropriate message for "limited functionality" or "unable to function" must be presented to the user.
- 5.9. In Canada, it is recommended best practice that the App should provide indication to the user to clearly disclose whether the App is sending the information off-board (i.e. cloud or mobile device). It is also recommended that there be a conspicuous visual indicator at all times during any data transmission. Each indicator should have a toggle feature that allows the user to discontinue data recording and data transmission.
- 5.10. Users must opt in prior to receiving direct marketing materials regarding the App.
- 5.11. All advertising made available in the App must comply with applicable law.
- 5.12. User must be given notice via the privacy statement of any intention to place advertising in or around the App before they download or activate the App.

5.13. User must be given the opportunity to opt out of advertising presented by the App, **except where the App's privacy statement specifies that advertising is required to obtain the services free or on a discounted basis.**

5.14. Advertising made available in the App that receives and stores user's behavior and/or preferences utilizing such tracking devices as cookies, web beacons, pixels or other technologies must adhere to the following restrictions:

5.14.1. Clear notice of behavioral tracking practices must be provided to users via the App's privacy statement (which must be presented to, and consented to, by the user). This notice must include:

- the types of information that will be collected for behavioral advertising purposes
- whether such information will be shared with an unaffiliated third party
- whether an App requires behavioral tracking in order for an App to run

5.14.2. Express consent of the user will be required if:

- a user's behavioral profile will be built on activity that occurs outside of the scope of the App (i.e. across multiple Apps or platforms)
- behavioral profiling or targeting will be carried out by a third party (i.e. ad network or mobile analytics company)

5.14.3. A user must be allowed the opportunity to opt out of all behavioral tracking within the App. This opt out may include the option for users to uninstall the App. If the App cannot facilitate an opt-out, then it cannot use behavioral tracking.

5.14.4. Information collected for behavioral purposes should only be retained as long as necessary to fulfill a legitimate business need, or as required by law.

5.14.5. Appropriate physical, electronic and administrative safeguards to protect the behavioral data should be maintained.

5.14.6. User's consent must be obtained prior to any material changes being made to the App's behavioral tracking policies.

5.14.7. User's consent must be obtained prior to collecting a user's personal information. Sensitive personal information such as financial account numbers, social security numbers or medical records should not be collected.

5.15. If the App or a "material" change to the App made by an update or upgrade were to do any of the following:

- provide enhanced user data collection or processing;
- interfere with the user's or an authorized user's control of the infotainment system;
- change or interfere with settings, preferences or commands already
- install or store on the infotainment system without the knowledge of the user or an authorized user of the infotainment system;

- change or interfere with data that is stored on the infotainment system in a manner that obstructs, interrupts or interferes with lawful access to or use of that data by the user or an authorized user of the infotainment system;
- cause the App or infotainment system to communicate with another computer system, or other device, without the authorization of the user or an authorized user of the App or infotainment system;
- install a computer program that may be activated by a third party without the knowledge of the user or an authorized user of the infotainment system; or
- perform a function that the developer knows and intends will cause the device to operate in a manner that is contrary to the reasonable expectations of the user (i.e. what was communicated to the user via the App's privacy statement or EULA)

then: a) prior notice must be provided to the user which clearly and prominently describes the nature and purpose of the upgrade or update and the upgrade or update's reasonably foreseeable impact on the operation of the App; and b) additional consent from the user must be obtained. In some jurisdictions (e.g., Canada), soliciting and obtaining such consent may require meeting specific form, content and acknowledgement requirements.

If a user does not consent to a material App update, the update cannot be implemented on that user's device.

6. App Requirements

6.1. General

6.1.1. The browser framework supports the following formats.

- HTML 5.0
- JavaScript 1.8
- CSS 2.0 / 3.0
- XML
- DOM
- XHTML
- XSLT and XPATH

6.1.2. App must not include or perform undocumented or hidden functions, or other functions that are contrary to the reasonable expectation of the user.

6.1.3. GM's expectation is to have one App for all the API levels.

6.1.4. Developer's App revisioning must use standard versioning major.minor.revision. Config parameter VERSION should be used by the App developer.

6.1.5. App must not have any inappropriate text or graphic (vulgar, rude, derogatory, etc.).

- 6.1.6. App must not violate or use any GM or other third party copyrights, trademarks, logos, or brand name. If any of these are used in the App, the Developer shall inform GM and GM has the right to verify that the Developer has the contractual rights to use them.
- 6.1.7. App must not be designed to circumvent laws or government regulations or facilitate unlawful activity of any kind.
- 6.1.8. Refer to the Developer Agreement for the GM APP SDK.
- 6.1.9. Developer Agreement for the GM APP SDK and Business agreements take precedence over the technical guidelines.

6.2. App Packaging Requirements

- 6.2.1. The App must be exported from the standard export function available in the GM SDK. An “App” consisting of multiple files, including HTML, JavaScript (JS), Cascading Style Sheets (CSS), and images (PNG, JPG, GIF) should be delivered to the Framework as one archive (e.g. zip) package. PHP or other server files need not be included in the zip file for *App* submission. App should reference any server files from their server location.
- 6.2.2. The infotainment system locally stores all files in the archive, which include HTML source files and static external resources. Dynamic data is not contained in the archive and is accessed through web service requests to an off-board data source.

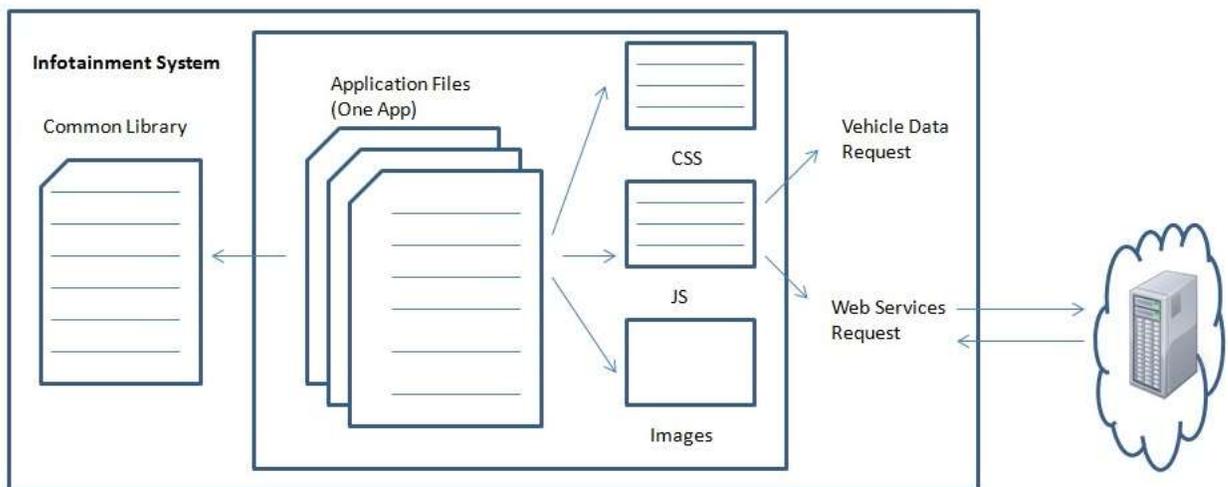


Figure: App package

6.3. Functional Requirements

6.3.1. About Screen and EULA

- 6.3.1.1. App must have an About screen that displays App Revision Number and GM's Version number. It must also have the App Name, a description of the functionality of the App and the Developers contact information (company name, mailing address and email address) . The GM function getVersionDisplayText must be used to get the GM version number.
- 6.3.1.2. It is solely your responsibility to ensure that your App will be made available under an end user license agreement ("EULA") between you and the end user that complies with all applicable laws in all countries where the App can be downloaded, installed, updated and/or upgraded. It is your sole responsibility to ensure that the EULA is substantially similar and does not conflict with the EULA specified in Appendix C. The EULA must describe the function and purpose of the App, set out your mailing and email address, and obtain whatever consent may be required to allow for the installation of the App, and to upgrade or update the App, on the user's device. You must also solicit and obtain whatever consent may be required to allow for the installation of the App, and to upgrade or update the App, on the user's device. If Your App includes an end user license agreement, You must provide a resource to allow for the end user to print a copy of such end user license agreement. In the event Your App does not include a valid end user license agreement, Your App shall be subject to the Standard End User License Agreement (which is part of the GM APP Catalog Terms and Conditions and included as Exhibit C to these Technical Guidelines for GM Infotainment App Developers).
- 6.3.1.3. The EULA [and the About screen] must be presented to each user in the vehicle. The user cannot use the app until they have agreed to the terms of the EULA [and seen the About screen]. Once the user agrees to the terms of the EULA, they need not be presented again.
- 6.3.1.4. The EULA agreed to by the user must be available via the About Screen.

6.3.2. Other

- 6.3.2.1. The following API's can only be executed based on direct user input (i.e. button press, Speech Rec, etc.)
 - dialPhoneNumber
 - sendTextMessage
 - setFavorite
 - sendEmail
 - startSpeechRecSession
- 6.3.2.2. Whenever an App uses the "showAlert" API, the alert must clearly identify what App it is from.

6.3.2.3. The “requestFocus” API is only used on user’s alert interaction, and only to a positive (i.e. primary) response. Note: For MY13, alert does not accept any response from the user, hence requestFocus has to be utilized only when absolutely needed. GM reserves the right to reject unnecessary requests to “requestFocus”.

6.3.2.4. When apps receive setShutDown callback, they have 3 seconds to complete gracefully. The framework executes the setShutDown callback which is registered during App startup via the GM API gm.system.init.

6.4. App Data Folder Requirements

6.4.1. Data is stored in a data folder (/data/share) for each App. An App must read or write data only to its associated location.

6.4.2. API call to write data will fail if the data folder is full. App must act appropriately when data folder is full.

6.4.3. Framework creates a folder for each user within the App’s data folder when an App is downloaded. App must write any private data into each user’s folder. Multiple users of the same App in the car must not see each other’s private data. When the App sets the “isprivate” flag attribute, data is written into App’s “data folder/user folder”.

6.4.4. App update replaces all the files in the Infotainment System that are in the updated App package. App is responsible to preserve user data in the data folder.

7. App Submission Requirements

7.1. Third party must complete and submit the App Submission Form (see “GM_Infotainment_App_Submission_Form.docx”) when they submit apps to GM.

7.2. GM needs the below information to post in the AppShop (for Buick, Chevrolet and GMC; different name for Cadillac).

7.2.1. App Icon Image (182x182 pixels)

7.2.2. App Contact for Customers. (String of 255 characters)

7.2.3. App Name or App Friendly Name.(String of 14 characters)

7.2.4. App Short Description.(String of 512 characters)

7.2.5. App Long Description.

7.2.6. Maximum of 5 screenshots (800x480) to be posted in the AppShop (for Buick, Chevrolet and GMC; different name for Cadillac).

7.2.7. Featured App Graphic (570x275)

Note: Please provide the descriptions in all the languages applicable to GM Infotainment platform, if App supports all those languages.

7.3. Any subsequent submission must have a version higher than the prior one.

8. Coding requirements

8.1. App must use only GM APIs specified in the SDK documentation.

- 8.2. App must not install or launch any OTHER executable code at all.
- 8.3. App must not download, link to, or accept any executable code.
- 8.4. App must not download or use any interpreted code unless prescribed by a GM API.
- 8.5. App must not have features or functions that are unlocked other than through the GM App Catalog and the requirements for the GM App Catalog.
- 8.6. App shall not act as a robot or spider, or link to or provide data to a robot or spider.
- 8.7. App must not use any off board JavaScript.
- 8.8. App must not include open source software that would require distribution or disclosure of any GM software.
- 8.9. App developer will be responsible for compliance with any direct and indirect obligations associated in any way with any Open Source Software that forms any part of the App. The obligations include, but are not limited to, compliance with all applicable licenses, copyrights, and any other legal obligations associated with the Open Source Software that is included in the App and with the distribution of the App and products containing the App.
- 8.10. Apps shall work reasonably with all the released versions of the API at the time of submission.
- 8.11. GM discourages the use of cookies and does not guarantee that they will work. Use of GM File I/O APIs is encouraged.
- 8.12. All Apps must use gm.system.init function at startup.
- 8.13. Apps must use gm.comm.webServiceRequest for any web calls for data.
- 8.14. App code must NOT be minified.
- 8.15. Apps must watch for Back Button Press and act appropriately. At the highest level of the App, the clearFocus API should be called.
- 8.16. Apps must not rely on using local storage in the HTML renderer. It may not be supported.
- 8.17. Apps must not include libraries that are not used by the App.

9. Communication Protocol

9.1. HTTP Connection

- The Framework supports HTTP and HTTPS requests over a TCP/IP transport layer.

9.2. Secure Connection

- The Framework will implement SSL and TLS in order to enable HTTPS connections.
- The infotainment system will accept and maintain public-key certificates necessary to support this protocol.

10. Security

- 10.1. App will be scanned for security vulnerabilities, malicious software and code quality metrics will be collected at the time of certification. The assessment methodology is based on emerging industry standards, such as those defined by

the Open Web App Security Project (OWASP), which include the OWASP Top 10 and App Security Verification Standard (ASVS).

- 10.2. Each App will be stored in a unique subdirectory of the root Internet App folder. App should access only local files within the same App directory.
- 10.3. The Infotainment system supports X.509 v3 root certificate store. The certificates pre-loaded in the store will be:

Comodo	Go Daddy	Thawte
DigiCert	IdenTrust	Trustwave
DigiNotar	Keynectis/Certplus	TURKTRUST
Entrust	Network Solutions	VeriSign
GeoTrust	QuoVadis	
GlobalSign	SwissSign	

- 10.4. The root certificate store will be utilized to enable a HTTPS connection.
- 10.5. Only the following Cipher suites are supported in the infotainment system.
 - Diffie Hellman Ephemeral key exchanges (DHE) (Anonymous DH is not permitted)
 - AES 128 or better
 - SHA 256
 - RSA 2048 or better
 - ECC with 256r1 curve or betterPer standard TLS communication, Individual Apps shall be able to specify which cipher suite to use with the server.
- 10.6. The Infotainment system does not accept self signed certificates.

11. Audio/Video Playback

- 11.1. Potential media frameworks that support decoding Internet audio and video streams include GStreamer and VLC player. The infotainment system will support streaming media (music, video, podcasts) of the following formats:

MP3	WMA	AAC and AAC+
PLS(not in MY13)	M3U	MP4
3GP	FLV	AVI
MPEG	H.263	M4a/b
H.264 (MY14 only)	RAM(MY14 only)	RMVB(MY14 only)
WEBM(MY14 only)		

- 11.2. The video codecs supported in the APIv1.0 of the framework is listed in “Video_Codecs_Supported_API_1.pdf”.
- 11.3. All the audio files in the App must sound reasonably clear.

- 11.4. Apps using Off-Board voice recognition must have reasonable in-vehicle performance. GM recommends using Nuance Voice Recognition Software.
- 11.5. Audio arbitration is managed by the framework. The callback function notifies the App of the arbitration status.

12. App Resource Utilization

- 12.1. App will be tested for resource utilization during certification process. App will not be certified if resource utilization (RAM footprint, CPU Load etc) is unreasonable.
- 12.2. Single images must not be larger than the screen size.

13.App Performance

- 13.1. App load time must be reasonable to be certified.

14.App fonts

- 14.1. The infotainment system will have a default font that Apps can use. However an App can include its own font too.

15.App Design, System Interaction and safety and validation requirements

- 15.1. Refer to User_Experience_Guidelines (also referred to as the HMI Guidelines) for App design, system interaction, and safety and validation requirements.
- 15.2. Apps are recommended to use GM widgets available in the GM SDK which are designed according to the HMI guidelines.
- 15.3. Apps must pass driver distraction testing in order to be certified.
- 15.4. App must have appropriate timeouts as mentioned in the HMI guidelines for all requests for internet content to avoid lockup and the user must be provided with an indication when timeouts occur.
- 15.5. Apps must watch for vehicle speed changes and adjust their App behavior to comply with the driver distraction guidelines section in the User_Experience_Guidelines document.
- 15.6. Apps must have a close button in the upper right hand corner.

16.App Usability

- 16.1. The App must not have any broken links.
- 16.2. The App must be tested to ensure that it does not crash or lock up.
- 16.3. App must act appropriately when it receives a negative response or error or no data from an API call request.
- 16.4. App must notify the user whenever large amounts of data are exchanged.

17. Example App Code Structure

This section describes the implementation of an example App. This App displays weather information for a given location. The following table illustrates the files that are necessary to implement this App and where the file would be stored.

Example File Structure & Descriptions – Weather App		
Filename	Description	Storage Location
index.html	Overall layout of the screen and location of all elements on the screen	module (WebApps/App10/index.html)
style.css	Format content contained in index.html	module (WebApps/App10/style.css)
weather.js	Adds animation to screen elements, controls event interactions, and updates screen elements	module (WebApps/App10/weather.js)
icons/	Directory of icons to be used	module (WebApps/App10/icons/)
weather.php	Receive requests from module, access weather sources (directly or through 3rd party API), and re-format data for GM App	Remote server

Files necessary to implement a basic weather App

18. App Configuration

18.1. Every App must have a config.xml at App root. This file is created when a new Dashboard project is created in the GM SDK. This file contains important information about the App which is used by the framework to download, configure and run the App.

18.2. The configuration parameters are listed below for.

XML Element Definitions			
Element	Value	Description	Data Usage
AppID	Unsigned Long	Unique ID of the App transferred. Uniqueness is managed by GM Back End. In-vehicle AppID shall match the GM Back End AppID. AppID is assigned	Uniquely identify an App. Handle sent to software for Favorites storage.

		by GM at the time of user id creation at the developers website.	
Version	Double	Version of the App transferred. Any subsequent submission must be a higher level than the prior one.	Check if an updated version is available for transfer.
AppType	Enum	This value specifies the type of data connection required by the App.	Based on cellular connection available, determine if the App can be used
VehicleStatus	Enum	This value specifies whether the App can be used while the vehicle is in motion or not.	Based on the speed and/or PRNDL value, determine if the App can be used
FriendlyName	String	User-readable App name. Must be less than 20 characters long.	Display with the icon on the home screen or favorites tab.
LocalPath	String	Relative path to store all files in the module that are transferred to the vehicle. The parent folder can be determined by module software, but the relative structure must remain intact. The LocalPath shall always be "App" + AppID. <i>Absolute Path:</i> <WebApps> / <LocalPath>	N/A
RemotePath	String	Relative path to the location of files for <AppID> that are stored on the GM Back End server.	N/A
IconURL	String	Filename of the image to be used as the App icon. <i>Remote Location:</i> <RemotePath> / <IconURL> <i>Local Location:</i> <WebApps> / <LocalPath> / <IconURL>	Display the App icon on the home screen or favorites tab.

MainURL	String	Filename of the initial file to show upon App launch. <i>Remote Location:</i> <RemotePath> / <MainURL> <i>Local Location:</i> <WebApps> / <LocalPath> / <MainURL>	When the App is launched, this URL should be rendered in the browser window. Therefore, this URL should be associated with the App on the home screen or favorites tab. If a hidden app, this is still the page which is launched to start execution.
AutoStart	Boolean	This value specifies whether the App should launch upon vehicle startup. When True, the App will automatically be launched as a background window upon completion of the network connection algorithm. When False, the App must be manually launched. GM may not approve the app to be AutoStart. Examples of apps to have AutoStart are Background Weather app or a Background Navigation app.	Start the app when the framework/module starts.
AppCategory	Enum	This value identifies the App category. The valid categories are: Audio, Monitoring, and General. This information is used to manage arbitration between multiple simultaneous Apps.	Classify each app so as to know when to close each app due to others of the same type being opened.
ReleaseNum	String	App Release Number. This can be anything chose by the app development team.	N/A

Table 1 XML configuration file schema

18.3. The structure of the config file is listed below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<gmWebApp>
  <AppID>02401019</AppID>
  <Version>0.1.0</AppVersion>
  <AppType>RemoteHigh</AppType>
  <VehicleStatus>Drive</VehicleStatus>
  <FriendlyName>AnotherTest</FriendlyName>
  <LocalPath>App02401019</LocalPath>
  <RemotePath>http://apps.gm.com/02401019</RemotePath>
  <IconURL>Icon.png</IconURL>
  <MainURL>index.html</MainURL>
  <AutoStart>>false</AutoStart>
  <AppCategory>General</AppCategory>
  <ReleaseNum>54</ReleaseNum>
</gmWebApp>
```

19. App Configuration Possible Values

19.1. AppId: Provided by GM when approved to create an App..

19.2. AppCategory: Audio, General.

19.3. Version: Standard Revisioning major.minor.releasenum.

19.4. AppType: Local, RemoteHigh, RemoteLow.

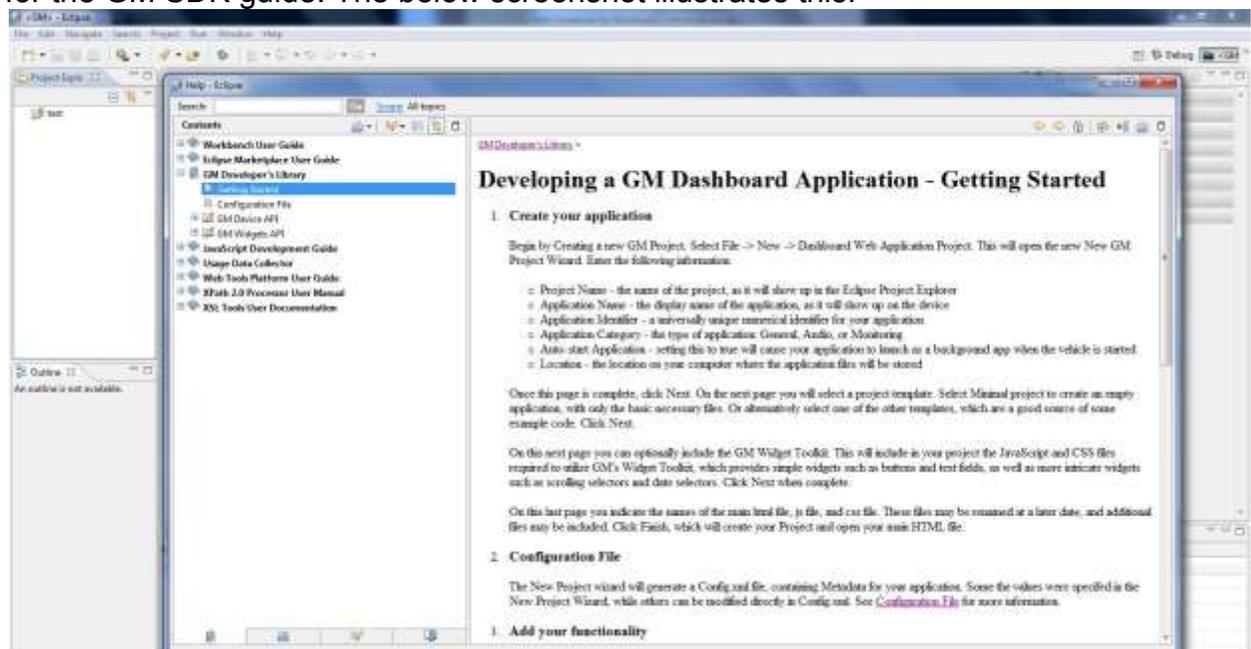
Local -> No data connection required

Remotelow -> Short bursts of data or nonstreaming data

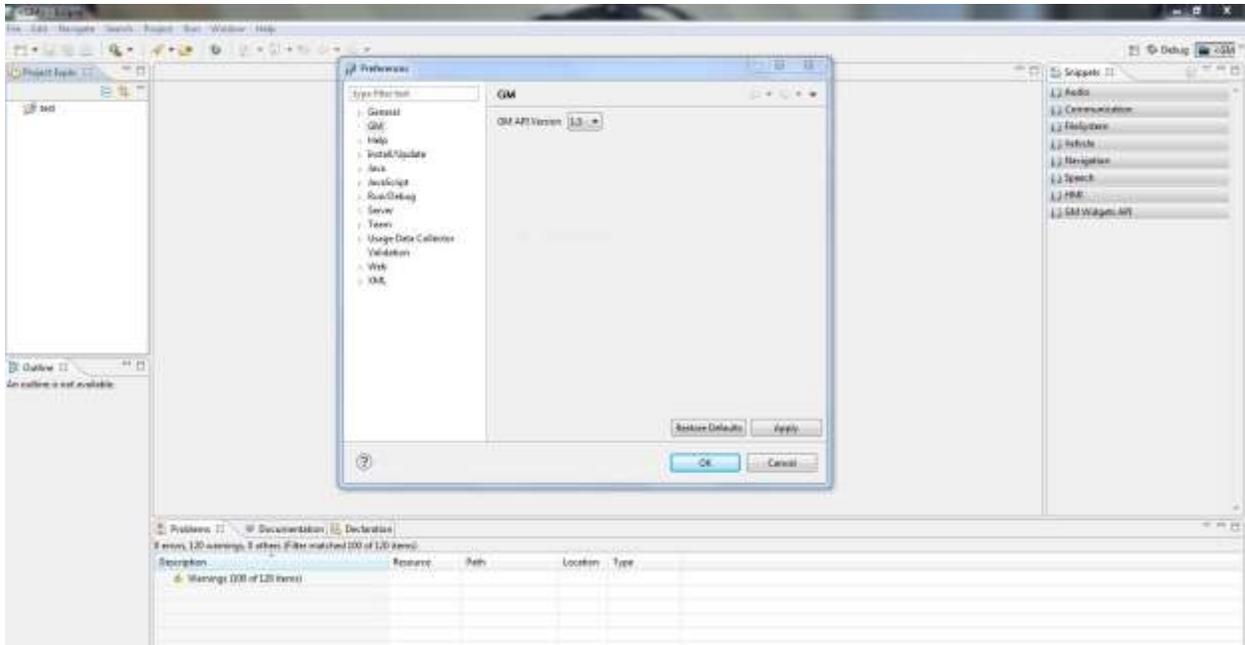
RemoteHigh -> Streaming data

20. GM SDK

20.1. Go to Eclipse and Click on Help Menu -> Help Contents -> GM Developer's library for the GM SDK guide. The below screenshot illustrates this.



20.2. API versions 1.0 and 1.5 are currently available in the GM SDK. The API setting can be changed in the SDK by clicking on Window->Preference->GM. The below screenshot illustrates this.



20.3. An SDK screencast video is available under the docs folder while downloading the SDK along with Install Instructions. The name of the file is gettingStartedVideo.html

21. GM SDK - Demo App / Sample Code

- 21.1. Go to eclipse and Click on File -> New -> Project.
- 21.2. Click on New Dashboard Web App under GM.
- 21.3. Fill in the project name and click Next.
- 21.4. Choose Code Samples Project and click Next.
- 21.5. Select the GM Widgets library and click Finish.

22. Living Document

This document represents GM's efforts to provide a helpful guide to develop and submit Apps to the Infotainment Platform and to share how Apps submitted to GM's Infotainment platform are reviewed. It is a living document that will continue to evolve and will be updated periodically to reflect the changes.

*The **Bluetooth**® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by General Motors Co. is under license. Other trademarks and trade names are those of their respective owners.*

