ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T F.749.8 (01/2024)

SERIES F: Non-telephone telecommunication services

Multimedia services

In-vehicle multimedia applets: Framework and functional requirements

ITU-T F-SERIES RECOMMENDATIONS

Non-telephone telecommunication services

TELEGRAPH SERVICE	F.1-F.109
Operating methods for the international public telegram service	F.1-F.19
The gentex network	F.20-F.29
Message switching	F.30-F.39
The international telemessage service	F.40-F.58
The international telex service	F.59-F.89
Statistics and publications on international telegraph services	F.90-F.99
Scheduled and leased communication services	F.100-F.104
Phototelegraph service	F.105-F.109
MOBILE SERVICE	F.110-F.159
Mobile services and multidestination satellite services	F.110-F.159
TELEMATIC SERVICES	F.160-F.399
Public facsimile service	F.160-F.199
Teletex service	F.200-F.299
Videotex service	F.300-F.349
General provisions for telematic services	F.350-F.399
MESSAGE HANDLING SERVICES	F.400-F.499
DIRECTORY SERVICES	F.500-F.549
DOCUMENT COMMUNICATION	F.550-F.599
Document communication	F.550-F.579
Programming communication interfaces	F.580-F.599
DATA TRANSMISSION SERVICES	F.600-F.699
MULTIMEDIA SERVICES	**F.700-F.799**
ISDN SERVICES	F.800-F.849
UNIVERSAL PERSONAL TELECOMMUNICATION	F.850-F.899
ACCESSIBILITY AND HUMAN FACTORS	F.900-F.999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T F.749.8

In-vehicle multimedia applets: Framework and functional requirements

Summary

Recommendation ITU-T F.749.8 describes the in-vehicle multimedia applet (VMMA) concept, the VMMA framework, the functional requirements, the functional APIs and the reference parameters.

Some detailed use cases and reference APIs are described in the appendix.

History *

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T F.749.8	2024-01-13	16	11.1002/1000/15762

Keywords

Application programming interface, in-vehicle multimedia applet, reference parameters.

* To access the Recommendation, type the URL https://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
3.1	Terms defined elsewhere	1
3.2	Terms defined in this Recommendation	2
4	Abbreviations and acronyms	2
5	Conventions	2
6	Overview of the VMMA	3
7	Reference framework of VMMA	3
8	Requirements	5
8.1	VMMA protocol requirements	5
8.2	VMMA basic interface requirements	5
8.3	Vehicle interface requirements	8
8.4	Navigation interface requirements	9
8.5	Speech interface requirements	10
8.6	Multimedia interface requirements	11
8.7	Display requirements	13
8.8	Performance monitoring requirements	13
Appendix I – Error codes		14
Appendix II – User interface examples		15
II.1	POST request	15
II.2	Login interface	15
II.3	User information interface	15
II.4	System information interface	16
II.5	Login expire interface	16
II.6	User setting interface	16
II.7	Vehicle speed interface	16
II.8	Vehicle fuel interface	16
II.9	VIN interface	16
II.10	Vehicle wheel rotation interface	17
II.11	Vehicle information interface	17
II.12	Navigation interface	17
II.13	Play TTS interface	17
II.14	Pause TTS interface	17
II.15	Resume TTS interface	17
II.16	TTS state change interface	17
II.17	Stop TTS interface	18

Page
II.18 Playlist interface 18
II.19 Display mode 18
Appendix III – Use case of in-vehicle multimedia applets 19
III.1 Navigation 19
III.2 Voice control 19
Bibliography 20

Recommendation ITU-T F.749.8

In-vehicle multimedia applets: Framework and functional requirements

1 Scope

This Recommendation describes the concept of in-vehicle multimedia applet (VMMA) common framework characteristics and defines functional requirements, then brings a capability framework, the application programming interfaces and the reference parameters of the VMMA.

The scope of this Recommendation includes:

– Overview of the VMMA;

– Reference framework of the VMMA;

– Functional requirements of the VMMA;

– Application programming interfaces and reference parameters of the VMMA.

In addition, the interface examples and the use cases of VMMA are provided in Appendices II and III, respectively.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.746.3] Recommendation ITU-T F.746.3 (2015), *Intelligent question answering service framework.*

[ITU-T H.625] Recommendation ITU-T H.625 (2017), *Architecture for network-based speech-to-speech translation services.*

[ITU-T P.10] Recommendation ITU-T P.10/G.100 (2017) *Vocabulary for performance, quality of service and quality of experience.*

[ECMA-262] ECMA-262 (2023), *ECMA Script 2023 Language Specification*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 vehicle multimedia system (VMS) [b-ITU-T F.749.3]: The VMS consists of vehicle multimedia system inputs (VM I/P), vehicle multimedia unit (VMU) and vehicle multimedia system outputs (VM O/P).

3.1.2 natural language processing (NLP) [ITU-T F.746.3]: A method that analyses text in natural languages through several processes such as part-of-speech recognition, syntactic analysis and semantic analysis.

3.1.3 text-to-speech synthesis (TTS) [ITU-T P.10]: A TTS process generates a speech signal from text codes. It is usually composed of the two parts:

– a language-dependent text-processing part (the high level processing part), which generates from the character string (by reading rules, vocabulary and semantic analysis) a set of phonetic, prosodic, etc., parameters which are used by:
– an acoustical signal generating part, the synthesizer itself, which generates the audible speech.

3.1.4 automatic speech recognition (ASR) [ITU-T H.625]: A system that can recognize continuous speech, often having phoneme-sized references, using lexical, syntactic, semantic, and pragmatic knowledge, and reacts appropriately (therefore having interpreted the message and found the corresponding action to be taken).

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 in-vehicle multimedia applet: In-vehicle multimedia applet (VMMA) is a new format of mobile application. VMMA not only provides a hybrid solution which relies on Web technologies (especially CSS and JavaScript) but also integrates with capabilities of native applications.

3.2.2 native applications: Software applications that are developed for a specific platform or operating system using the native programming languages and tools provided by that platform.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ASR Automatic Speech Recognition
CSS Cascading Style Sheets
HMI Human–Machine Interface
LAN Local Area Network
LBS Location-Based Service
NLP Natural Language Processing
SDK Software Development Kit
TTS Text-to-Speech
TTS Text-To-Speech
UI User Interface
URL Uniform Resource Locator
VIN Vehicle Identification Number
VM I/P Vehicle Multimedia System Inputs
VM O/P Vehicle Multimedia System Outputs
VMMA In-Vehicle Multimedia Applet
VMS Vehicle Multimedia System
VMU Vehicle Multimedia Unit

5 Conventions

In this Recommendation:

– The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.
– The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.
– The keyword "**may**" indicates an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

In the tables with requirements, optionality is denoted as follows: M for mandatory, O for optional.

6 Overview of the VMMA

The in-vehicle multimedia applet (VMMA) is a new form of in-car application, leveraging both Web technologies (especially CSS and JavaScript [ECMAScript]) as well as capabilities of native applications.

Without installation, the VMMA has a dynamic loading and update mechanism and can be greatly convenient for users to use for entertainment content. Combined with the intelligent voice system, the in-vehicle application will search for contents or services seamlessly. The whole process supports voice interaction, which effectively improves the application interaction experience in the vehicle.

The VMMA enhances the service ecosystem in-vehicle, combines the scene perception, location-based service (LBS), social media etc., and provides more secure and nature scenarized services for the user. VMMA also provides uniform APIs and protocols compatible with different vehicle systems and hardware platforms.

7 Reference framework of VMMA

The VMMA is designed to be integrated into a vehicle's multimedia system. It offers a range of features and functionalities to enhance the driving experience while providing seamless connectivity with other infotainment services and platforms.

The VMMA does not need to be installed. And the mechanism of dynamic loading and updating can greatly facilitate the user's behaviour and content consumption. Combined with the intelligent voice interaction in the whole processes, the VMMA can effectively improve the application interaction experience in the car.

Figure 1 shows the reference framework of VMMA.

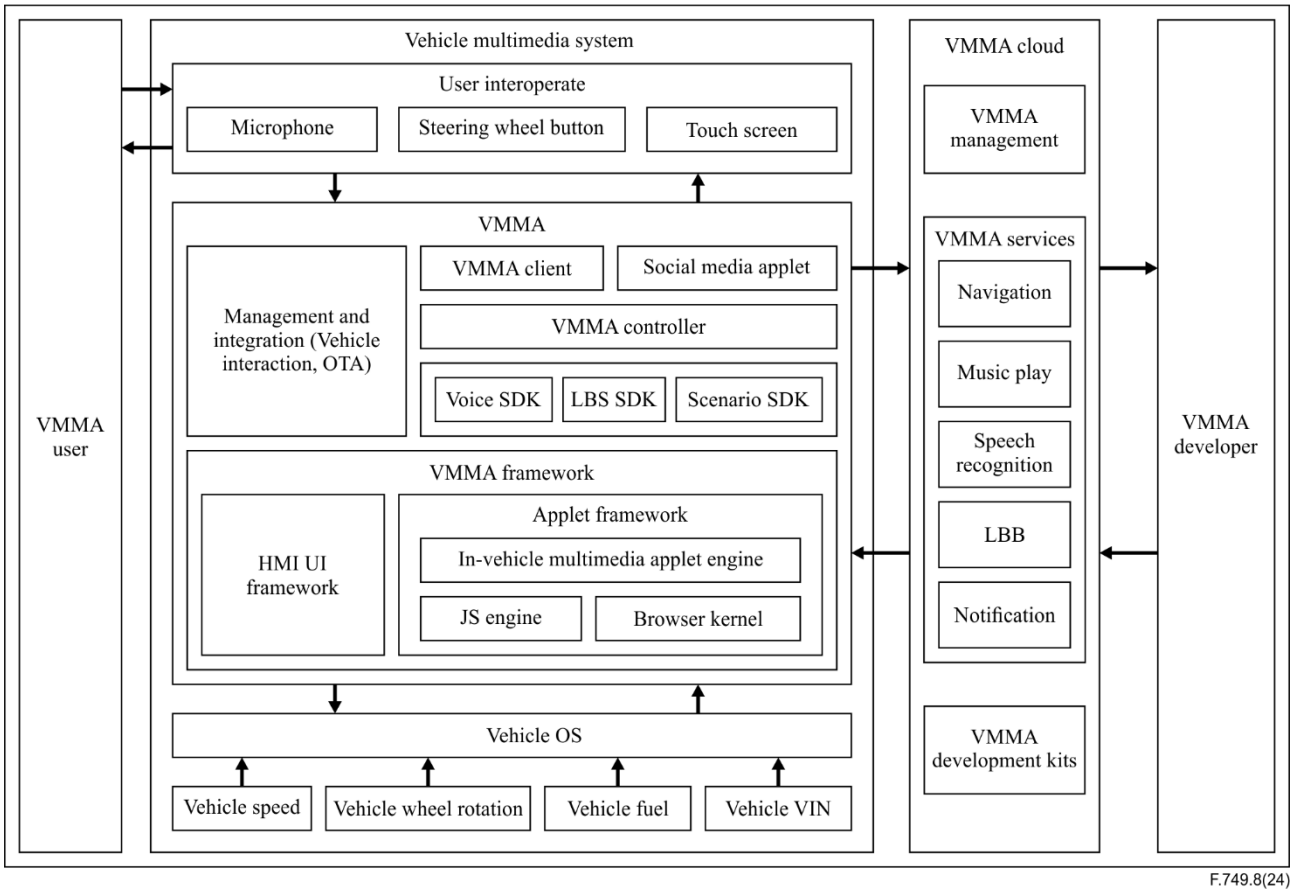

Figure 1 – Reference framework of VMMA

The framework of VMMA is composed of the user, the user interoperate module, the VMMA function module, the vehicle multimedia system, the VMMA cloud and the VMMA developer.

The users can control VMMA through voice, steering wheel buttons and touch screens. The VMMA is recommended to be compatible with traditional vehicle interactive control modes.

The VMMA is recommended to be based on the JavaScript framework, to be compatible with social media applets, and to include HMI UI framework and applet framework. The VMMA management module is required to upgrade configurations. The VMMA controller is required to support a variety of speech recognition technologies such as automatic speech recognition (ASR) [ITU-T H.625], text-to-speech synthesis (TTS) [ITU-T P.10] and natural language processing (NLP) [ITU-T F.746.3]. It is also required to support voice control SDK, location-based services software development kit (LBS SDK), and scenario SDK. A voice control SDK is a set of tools, libraries and documentation that enables developers to create, test and deploy applications or features that incorporate voice recognition, voice synthesis and other voice-related technologies. An LBS SDK is a collection of tools, libraries and documentation that enables developers to create, test and deploy applications or features that utilize location-based services. These services typically involve the use of GPS, wireless LAN ([b-IEEE 802.11a], [b-IEEE 802.11b], [b-IEEE 802.11g], [b-IEEE 802.11n], [b-IEEE 802.11ac], [b-IEEE 802.11ax]) or cellular network data to provide real-time location information, allowing developers to create location-aware applications, such as navigation, geofencing and location-based marketing or analytics. A scenario SDK is a set of tools, libraries and documentation that enable developers to create, test, and deploy scenario-based applications or features for specific platforms or environments.

The VMMA framework is required to have an HMI UI framework, a JavaScript engine and a web browser kernel. A JavaScript engine is a software program or interpreter that processes and executes JavaScript code in web browsers. It translates the human-readable JavaScript code into machine-

readable instructions, enabling the execution of interactive VMMA, and other JavaScript-based functionalities.

The VMMA cloud is composed of the VMMA management platform, the VMMA services and the VMMA development kits. The VMMA management provides monitoring, operations and management functions of VMMA, including VMMA uploading, distribution and updating. The VMMA services provide the navigation, music play, speech recognition, LBS and notification functions. The VMMA development kits provide the development tool kits for VMMA developers.

8 Requirements

8.1 VMMA protocol requirements

[PRO-001] The protocol of the vehicle applet is required to use HTTPS POST request with "Content-type: application/json" and "Content-encoding: UTF-8", and the request is returned as a Json object.

The VMMA protocol parameters are shown in Table 1.

Table 1 – VMMA protocol parameter

Element/attribute	Description	Necessity
appid	Applet id	M
sign	Signature	M
timestamp	Time stamp, second	M
nonce	Random string	M
code	Execution code	M
version	Protocol version	M
grant_type	Authorization code	O

8.2 VMMA basic interface requirements

[BIR-001] The VMMA basic interface is required to have login interface, user information interface, system information interface, login expiration interface and user setting interface.

8.2.1 Login

User login interface is shown as below, and the parameters are shown in Table 2.

```
login(Object object)
```

Table 2 – Login parameter

Element/attribute	Type	Description	Necessity
timeout	number	Time out, millisecond	O
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

The login callback parameters are shown in Table 3.

Table 3 – Login callback parameter

Element/attribute	Type	Description
code	string	User login credentials

8.2.2 User information interface

The getUserInfo object is used to retrieve the user information.

The interface is shown below, and the parameters are shown in Table 4.

```
getUserInfo (Object object)
```

Table 4 – User information parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O
withCredentials	boolean	When withCredentials is true, it requires that login has previously been called and that the login status has not expired. When withCredentials is false, login status is not required.	O

The user information callback parameters are shown in Table 5.

Table 5 – User information callback parameter

Element/attribute	Type	Description
userInfo	number	The user information object does not contain sensitive information such as openid
rawData	number	The raw data string, excluding sensitive information, is used to calculate the signature
msignature	number	Used to verify user information
encryptedData	number	Encrypted data of complete user information
iv	number	The initial vector of the encryption algorithm

8.2.3 System information interface

The getSystemInfo object is used to retrieve the vehicle system information.

The interface is shown below, and the parameters are shown in Table 6.

```
getSystemInfo (Object object)
```

Table 6 – System information parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

The get system information interface callback parameters are shown in Table 7.

Table 7 – System information callback parameter

Element/attribute	Type	Description
brand	string	Vehicle brand
model	string	Smartphone model
pixelRatio	number	Pixel ratio
screenWidth	number	Screen width
screenHeight	number	Screen height
windowWidth	number	Window width
windowHeight	number	Window height
language	string	System language
version	string	Version
system	string	Operation system
platform	string	Smartphone operation system
SDKVersion	string	SDK version
orientation	string	Screen display status: landscape-full, landscape-half, portrait-full, portrait-half
device	string	Device type

8.2.4 Login expiration interface

The checkSession object is used to check if the login status has expired.

The interface is shown below, and the parameters are shown in Table 8.

```
checkSession(Object object)
```

Table 8 – Login expire parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.2.5 User setting interface

The getSetting object is used to retrieve the user's authorization settings.

The interface is shown below, and the parameters are shown in Table 9.

```
getSetting(Object object)
```

Table 9 – User setting parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

The get user setting callback parameters are shown in Table 10.

Table 10 – User setting callback parameter

Element/Attribute	Type	Description
authSetting	string	Return user authorization result

8.3 Vehicle interface requirements

[VIR-001] The vehicle interface is required to have vehicle speed interface, vehicle fuel interface, vehicle identification number (VIN) interface, vehicle wheel rotation interface and vehicle information interface.

8.3.1 Vehicle speed interface

The getSpeed object is used to retrieve the vehicle speed.

The interface is shown below and the parameters are shown in Table 11.

```
getSpeed(Object object)
```

Table 11 – Vehicle speed parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.3.2 Vehicle fuel interface

The getRemainPower object is used to retrieve the vehicle fuel.

The interface is shown below, and the parameters are shown in Table 12.

```
getRemainPower(Object object)
```

Table 12 – Vehicle fuel parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.3.3 Vehicle VIN interface

The getVehicleId object is used to retrieve the VIN.

The interface is shown below and the parameters are shown in Table 13.

```
getVehicleId(Object object)
```

Table 13 – Vehicle VIN parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.3.4 Vehicle wheel rotation interface

The getWheelRotationAngle object is used to retrieve the vehicle wheel rotation.

The interface is shown below, and the parameters are shown in Table 14.

```
getWheelRotationAngle(Object object)
```

Table 14 – Vehicle wheel rotation parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.3.5 Vehicle information interface

The addVehicleEventListener object is used to retrieve the vehicle authorization information.

The interface is shown below and the parameters are shown in Table 15.

```
addVehicleEventListener(Object object, Function callback)
```

Table 15 – Vehicle information parameter

Element/attribute	Type	Description
speed	array	Get speed change
keyCode	array	Get keycode change

8.4 Navigation interface requirements

[NIR-001] The navigation interface is required to have the interface to provide the navigation map.

The navigation interface is shown below, and the parameters are shown in Table 16.

```
navigateMap(Object object)
```

Table 16 – Map parameter

Element/attribute	Type	Description
destination	object	Destination

The destination parameters are shown in Table 17.

Table 17 – Destination parameter

Element/attribute	Type	Description	Necessity
latitude	number	Latitude	M
longitude	number	Longitude	M
address	string	Address	M
name	string	Name	O

8.5 Speech interface requirements

[SIR-001] The speech interface is required to have play TTS interface, pause TTS interface, resume TTS interface, TTS state change interface and stop TTS interface.

8.5.1 Play TTS interface

The play TTS interface is shown below, and the parameters are shown in Table 18.

```
playTTS(Object object)
```

Table 18 – PlayTTS parameter

Element/attribute	Type	Description	Necessity
connect	string	Play TTS	M
playAsMedia	boolean	True for long TTS broadcast False for short TTS broadcast	O
volume	number	Volume	O
success	string	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.5.2 Pause TTS interface

The pause TTS interface is shown below, and the parameters are shown in Table 19.

```
pauseTTS(Object object)
```

Table 19 – Pause TTS parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.5.3 Resume TTS interface

The resume TTS interface is shown below, and the parameters are shown in Table 20.

```
resumeTTS(Object object)
```

Table 20 – Resume TTS parameter

Element/attribute	Type	Description	Necessity
success	function	Return success	O
fail	function	Return fail	O
complete	function	Return complete	O

8.5.4 TTS state change interface

The TTS state change interface is shown below, and the parameters are shown in Table 21.

```
onTTSStateChange(Function callback)
```

Table 21 – TTS state change callback parameter

Element/attribute	Type	Description
state	string	'playing' 'paused' 'stopped' 'complete' 'error' 'pretrack' 'nexttrack'

8.5.5 Stop TTS interface

The stop TTS interface is shown below and the parameters are shown in Table 22.

```
stopTTS(Object object)
```

Table 22 – Stop TTS parameter

Element/attribute	Type	Description	Necessity
msgid	string	Return playTTS	M

8.6 Multimedia interface requirements

[MIR-001] The multimedia interface is required to have play music interface, get playlist interface and music play process interface.

8.6.1 Play music interface

The play music interface is shown below, and the parameters are shown in Table 23.

```
playmusic(Object object)
```

Table 23 – Music parameter

Element/attribute	Description	Necessity
operationType	"play","pause","stop","resume","seek"	M
src	Multimedia source	M
title	Multimedia title	M
coverImgUrl	Cover image URL	M

Table 23 – Music parameter

Element/attribute	Description	Necessity
singer	Singer name	M
epname	Album name	M
currentTime	Time and location of media content（second）	M
state	"pause","stop","play","timeupdate","waiting", "error", "occupied", "pretrack","nexttrack"	M
errCode	Error code (Appendix I)	O
errMsg	Error message	O
paused	Paused	O
duration	Multimedia duration	O
currentTime	Current time	O
buffered	Buffer	O

8.6.2 Get playlist interface

The get playlist interface is shown below, and the parameters are shown in Table 24.

```
getplaylist(Object object)
```

Table 24 – Playlist parameter

Element/attribute	Type	Description	Necessity
title	string	Media content title	M
singer	string	Singer	M
coverImgUrl	string	Album URL	O
dataUrl	string	Media content source URL	M
options	string	Custom fields used by developers to store custom information describing the current album	O

8.6.3 Music play process interface

The message sequence among Vehicle OS, VMMA framework and VMMA client in-vehicle are shown in Figure 2.

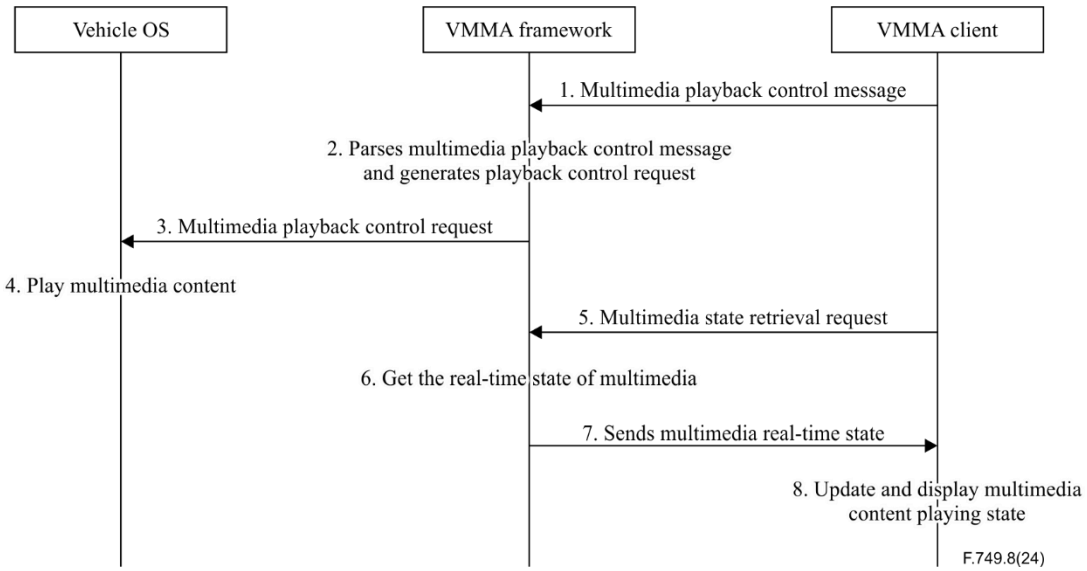

Figure 2 – The VMMA message sequence for music play in-vehicle

1. The VMMA client sends the music playback control message to the VMMA framework.
2. The VMMA framework parses the music playback control message and generates the music playback control request.
3. The VMMA framework sends the music playback control request to the vehicle operating system (OS).
4. The vehicle OS plays and controls the music according to the music control request.
5. The VMMA client sends a music state retrieval request to the VMMA framework.
6. The VMMA framework obtains the real-time status of music according to the music state retrieval request.
7. The VMMA framework sends the real-time state of music to the VMMA client.
8. The VMMA client updates and displays the multimedia real-time state.

8.7 Display requirements

[DIS-001] The VMMA is required to ensure the width-to-height ratio of the visible area.

NOTE – The car head unit screen has a variety of sizes, and the screen orientation is divided into horizontal screen and vertical screen.

8.8 Performance monitoring requirements

[PMC-001] The VMMA is required to support the monitoring functions for the start-up time, memory usage and open failure rate statistics.

Appendix I

Error codes

(This appendix does not form an integral part of this Recommendation.)

Table I.1 shows the error code returned when the VMMA runs abnormally.

Table I.1 – Error codes

errCode(string)	**Description**
−404	Account error
−403	Need to login
−402	Illegal permissions
−401	Denied authorization
−1	Service error
3	Parameter error
5	Authorization error
101	User ID does not exist
104	Invalid package
153	Invalid session key
250	Car id does not exist
317	Logged in
321	Not login
408	Request timeout

Appendix II

User interface examples

(This appendix does not form an integral part of this Recommendation.)

Clauses II.1 to II.19 show the user interface examples.

II.1 POST request

```
POST request:
{
     "appid":"60258",
     "nonce":"abcde6789",
     "timestamp":"1606286021",
     "version":"1.0",
     "code":"29225ba02ee811eba252b55ce8cf45c8",
     "grant_type":"authorization_code",
     "sign":"626802ae19bd45a33855594b3d2450df"
}
```

II.2 Login interface

```
login({
 success (res) {
 if (res.code) {
 wx.request({
 url: 'https://example.com/onLogin',
 data: {
 code: res.code
 }
 })
 } else {
 console.log('Login Failed' + res.errMsg)
 }
 }
})
```

II.3 User information interface

```
getUserInfo({
 success: function(res) {
 var userInfo = res.userInfo
 var nickName = userInfo.nickName
 var avatarUrl = userInfo.avatarUrl
 var gender = userInfo.gender
 var province = userInfo.province
 var city = userInfo.city
 var country = userInfo.country
 }
})
encryptedData
{
 "openId": "OPENID",
 "nickName": "NICKNAME",
 "gender": GENDER,
 "city": "CITY",
 "province": "PROVINCE",
 "country": "COUNTRY",
 "avatarUrl": "AVATARURL",
 "watermark": {
 "appid": "APPID",
```

```
 "timestamp": TIMESTAMP
 }
}
```

II.4 System information interface

```
getSystemInfo({
success (res) {
console.log(res.device)
}
})
try {
 const res = wx.getSystemInfoSync()
 console.log(res.device)
} catch (e) {

}
```

II.5 Login expire interface

```
checkSession({
 success () {
 //session_key
 },
 fail () {
 // session_key
 login()
 }
})
```

II.6 User setting interface

```
getSetting({
success (res) {
console.log(res.authSetting)
res.authSetting = {
"scope.userInfo": true,
"scope.userLocation": true
 }
}
})
```

II.7 Vehicle speed interface

```
getSpeed({
 success: function(res) {
 console.log('get vehicle speed', res);
 }
})
```

II.8 Vehicle fuel interface

```
getRemainPower({
 success: function(res) {
 console.log('get vehicle fuel', res);
 }
})
```

II.9 VIN interface

```
getVechicleId({
 success: function(res) {
 console.log('get vehicle vin', res);
 }
})
```

II.10 Vehicle wheel rotation interface

```
getWheelRotationAngle({
 success: function(res) {
 console.log('get vehicle wheel rotation', res);
 }
})
```

II.11 Vehicle information interface

```
addVehicleEventListener({keyCode: ['keycode'],speed: ['speed']},
function(data) {
data.forEach(function(item){
 if (item.hasOwnProperty('speed')) {
 console.log('vehicle speed',item.speed)
 }
 if (item.hasOwnProperty('keycode')) {
 console.log('keycode',item.keycode)
 }
 })
})
```

II.12 Navigation interface

```
const tamLatitude = 39.908692;
const tamLongitude = 116.397433;
const tamAddress = "XXXX";

navigateMap({
 destination: {
 latitude: tamLatitude,
 longitude: tamLongitude,
 address: "xxxx"
 }
})
```

II.13 Play TTS interface

```
playTTS({
 content: 'find the gas station',
 volume: 50,
 success (res) { }
})
```

II.14 Pause TTS interface

```
pauseTTS({
 success (res) { }
})
```

II.15 Resume TTS interface

```
resumeTTS({
 success (res) { }
})
```

II.16 TTS state change interface

```
onTTSStateChange(function(res) {
 console.log(JSON.stringify(res.state));
 // Output:  {"state":"COMPLETE"}
})
```

II.17 Stop TTS interface

```
stopTTS({
 msgId: this.data.msgId,
 success: function (res) {
 console.log(res)
 })
```

II.18 Playlist interface

```
var curPlayList = [{

coverImgUrl:'http://image.biaobaiju.com/uploads/20180803/23/153
3309822-GCcDphRmqw.jpg',
 title: 'xxxx1',
 singer: 'xxxx1',
 dataUrl:
'http://music.163.com/song/media/outer/url?id=476592630.mp3',
 options: JSON.stringify({
 epname: 'xxxx'
 })
},
{
 coverImgUrl:
'https://qpic.y.qq.com/music_cover/ezXpob9biaedyoUWFJDttEQc9Hsq
SUpIiaPVnKrwkl94QrXna9xZ5wbg/300?n=1',
 title: 'xxxx2',
 singer: 'xxxx2',
 dataUrl: 'https://cdn.kaishuhezi.com/audio/story/M-2-01-
20150408.mp3'
},
{
 coverImgUrl:
'https://qpic.y.qq.com/music_cover/HvzXRphXYwDIM7tpX0XM4dCF2K6t
u1k38ts4H8sD5icicTtoeZPsM09g/300?n=1',
 title: 'xxxx3',
 singer: 'xxxx3',
 dataUrl: 'https://cdn.kaishuhezi.com/audio/story/M-1-01-
20150408.mp3'
}
];
const audioPlayer = wx.getBackgroundAudioManager()
audioPlayer.playInfo = {
playList:curPlayList,
context: JSON.stringify({key:1000,timeStamp:536760})
}
audioPlayer.src =
"http://music.163.com/song/media/outer/url?id=476592630.mp3";
```

II.19 Display mode

```
<br/>
  if (window.width > window.height) {
    return 'horizontal'
  } else {
    return 'portrait'
  }
<br/>
```

Appendix III

Use case of in-vehicle multimedia applets

(This appendix does not form an integral part of this Recommendation.)

III.1 Navigation

Distribution of in-vehicle multimedia applet is based on LBS geofence.

The VMMA could support home page, search, navigation and map functions. The VMMA could reply to a real-time traffic information query and points of interest query for navigation.

III.2 Voice control

Voice control of VMMA supports voice switch, voice control page up and down, and page return.

Voice control of VMMA containing multimedia players supports play, pause, up and down, fast forward, fast back and other operations.

Bibliography

[b-ITU-T F.749.3] Recommendation ITU-T F.749.3 (2020), *Use cases and requirements for vehicular multimedia networks.*

[b-IEEE 802.11a] Standard, IEEE 802.11a (1999), *Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless medium access control (MAC) and physical layer (PHY) specifications: High speed physical layer in the 5 GHz band*.

[b-IEEE 802.11b] Standard, IEEE 802.11b (2001), *Telecommunications and information exchange between systems – Local and Metropolitan networks – Specific requirements – Part 11. Cor-1: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher speed physical layer (PHY) extension in the 2.4 GHz band – corrigendum 1*.

[b-IEEE 802.11g] Standard, IEEE 802.11g (2003), *Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Further higher data rate extension in the 2.4 GHz band*.

[b-IEEE 802.11n] Standard, IEEE 802.11n (2009), *Information technology – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN medium access control (MAC)and physical layer (PHY) specifications Amendment 5: Enhancements for higher throughput*.

[b-IEEE 802.11ac] Standard, IEEE 802.11ac (2013), *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks--Specific requirements – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications – amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz.*

[b-IEEE 802.11ax] Standard, IEEE 802.11ax (2021), *Information Technology – Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks – Specific Requirements Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications – Amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	**Non-telephone telecommunication services**
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2024