



Grandstream Networks, Inc.

GXV3175 IP Multimedia Phone

GMI HTML SDK and API Guide



1 GMI V2.0 Overview

GMI (Grandstream Manager Interface) is a management API developed by Grandstream Networks. Designed for our IP Multimedia phones, it allows partners to develop customized applications on the phone.

GMI supports standard HTML/CSS/Javascript, users can use these dynamic web page development languages to develop their customized application. GMI will display the application on the phone based on the web pages generated.

Additionally, GMI provides several basic API functions (packaged in Javascript) to facilitate users calling the existing applications on the multimedia phone or to obtain the phone status etc. This allows users with basic web application programming skills to develop their customized application on the multimedia phone, without the effort of learning a new programming language. All that is required for the user is to understand how the GMI works and how to use these simple APIs to interact with the phone.

GXV3175 supports GMI Verion 2.0. GMI V2.0 has two types of JavaScript interface:

- simpleGMI: Provides a few simple and fast interfaces. It is highly efficient but has no return value. Users could not obtain execution result nor interface for specific data.
- GMIEngine: Provides return value(s) and enables users to obtain interface for specific data and environment variables.

2 Remote GMI application

To add a GMI application to the multimedia phone, users need to add the following highlighted lines of code in *menu.xml* (in [GXV3175 XML GUI Customization Tool](#) package):

```
<submenu func-name="tools">
  <item func-name="ip2location"/>
  <item func-name="filemanager"/>
  <item func-name="calendar"/>
  <item func-name="calculator"/>
  <item func-name="alarmclock"/>
  <link icon="icon/Grandstream.png" display-name="GMI" type="gmiapp">
    <link-url>http://www.ipvideotalk.com/V2/</link-url>
  </link>
</submenu>
```

The highlighted lines of code will add a menu selection "GMI" under your phone's "Tools"

submenu. The parameter “icon” specifies the icon displayed and “display-name” specifies the text displayed in the phone menu. The parameter type=“gmiapp” specifies that this is a GMI application, and the parameter “link-url” specifies the URL address at which this application is stored. In this example, the phone will point to the link below to get the corresponding application to display on the phone: <http://www.ipvideotalk.com/V2/> . Users can modify parameters “icon”, “link-url” and “display-name” to suit their needs.

The above example files are available for download from:

www.grandstream.com/products/gxv_series/gxv3175/resources/gxv3175_gmi_examples/Demo_V2.zip.

Users will then need to use the [GXV3175 XML GUI Customization Tool](#) to generate a custom file “gxv3170cust” and upload it to the phone via the firmware upgrade process. For more information regarding GXV3175 XML GUI Customization, please refer to the following document: [GXV31XX XML GUI Customization Guide](#).

3 Local GMI application

Users may prefer local GMI application when the network speed is a concern for remote access. Please follow the procedures below to add a local GMI application to GXV3175.

- Download the following GMI example package from Grandstream web site:
www.grandstream.com/products/gxv_series/gxv3175/resources/gxv3175_gmi_examples/Hotel.zip
- Unzip and copy it to GXV3175 with a storage device. The main file *index.html* will be located under */sda1/Hotel* directory. (Depending on the storage device type, the folder name in the File Manager could be sda1, udisk, sda2 etc).
- Add the following highlighted content in *menu.xml*:

```
<submenu func-name="tools">
  <item func-name="ip2location"/>
  <item func-name="filemanager"/>
  <item func-name="calendar"/>
  <item func-name="calculator"/>
  <item func-name="alarmclock"/>
  <link icon="icon/Grandstream.png" display-name="GMI" type="gmiapp">
    <link-url>file:///sda1/Hotel/index.html</link-url>
  </link>
</submenu>
```
- Use the GXV3175 XML GUI Customization Tool to generate the “gxv3170cust” file with the updated *menu.xml* file.
- Load the file “gxv3170cust” onto GXV3175 via firmware upgrade process.
- Reboot the GXV3175, a local GMI application will be added to GXV3175.

4 simpleGMI Interface

4.1 simpleGMI.refresh()

Function	simpleGMI.refresh()
Description	Refresh the current page. The phone will obtain the current page from the web server and reload the page on the phone.
Parameters and Return value	Parameter: N/A Return value: N/A
Note	This interface is mainly for debugging. It is recommended to remove this function in your program before official release so that the program will run as smoothly as the built-in applications on the phone, providing users with better user experience.

4.2 simpleGMI.historypage()

Function	simpleGMI.historypage(num)
Description	Go to the page visited in history, as specified in parameter num.
Parameters and Return value	Parameter: num - Any natural number. - Negative number represents the number of page records to go backward; 0 represents the current page; - Positive number represents the number of page records to go forward. For example, the user visited these web pages in the following order: www.google.com www.yahoo.com

	www.baidu.com When the user is browsing the webpage www.baidu.com, the function call to: simpleGMI.historypage(-1) will allow the user to return to the last page visited (www.yahoo.com). At this point, the last page visited will be www.google.com. If the function: simpleGMI.historypage(-1) is called again, the user will return to www.google.com and the next page visited is www.baidu.com. If the function: simpleGMI.historypage(1) is called, the user will return to www.yahoo.com. If num is a non-existent record, nothing will take effect. Return Value: N/A
Note	

4.3 simpleGMI.gotoURL()

Function	simpleGMI.gotoURL(url)
Description	Go to the URL address specified in the url parameter.
Parameters and Return value	Parameter: url - any url Return Value: N/A
Note	This function is used to implement a hyperlink.

4.4 simpleGMI.dial()

Function	simpleGMI.dial(acct, isVideo, isDialPlan, number, headers)
Description	Call a specified number from a specified account.

Parameters and Return value	Parameter: acct - the account to be used starting from 0 isVideo - video call or not. 1-Yes, 0-No isDialPlan - check dial plan or not. 1-Yes, 0-No number - number to be dialed headers - add in SIP header. For instance, when "headers" is: myheader1=myvalue1&myheader2=myvalue2 When Call is initiated, the following two headers will be added in INVITE: myheader1: myvalue1 myheader2: myvalue2 Return Value: N/A
Note	N/A

4.5 simpleGMI.transfer()

Function	simpleGMI.transfer()
Description	Go to Transfer status.
Parameter and Return Value	Parameter: N/A Return Value: N/A
Note	Valid only when the current line is connected in a call.

4.6 simpleGMI.transferTo()

Function	simpleGMI.transferTo(destnum)
Description	In transfer status, transfer the current call to destination number.
Parameter and Return Value	Parameter:

	destnum - the destination number to be transferred to Return Value: N/A
Note	Valid only when the current call is in transfer status. If users would like to use one-key-transfer, users could enter the number first then call simpleGMI.transfer() and simpleGMI.simpleGMI(destnum) in the return function to dial out.

4.7 SimpleGMI.hangup()

Function	SimpleGMI.hangup()
Description	Hang up the current call.
Parameter and Return Value	Parameter: N/A Return Value: N/A
Note	N/A

4.8 simpleGMI.launchService()

Function	simpleGMI.launchService(program)
Description	Launch the specified program.
Parameter and Return Value	Parameter: program - specify the program to be launched The program (displayed as the menu item name on the phone) to be launched and the corresponding code which could be sent to launch interface as parameter are as follows: Contacts - "Phonebook" Call History - "CallHistory" Message - "Message" Browser - "Browser"

	AlarmClock - "AlarmClock"
	BunnyHunt - "BunnyHunt"
	Picture Matching - "Buzzy"
	Calculator - "Calculator"
	Calendar - "Calendar"
	Color Code - "colorcode"
	Facebook - "Facebook"
	File Manager - "FileManager"
	Flickr - "Flickr"
	Gobang - "Gobang"
	Google Voice - "GoogleVoice"
	Tetris - "gottet"
	Internet Radio - "InternetRadio"
	IP to Location - "IP2Location"
	Last.FM - "LastFM"
	Media Player - "MediaPlayer"
	Movie Trailer - "MovieTrailer"
	News Videos - "NewsVideos"
	Peg - "peg"
	Phanfare - "Phanfare"
	Photobucket - "Photobucket"
	Camara - "PIP"
	Checker - "Qchecker"
	Sudoku - "simsu"
	Slide Show - "SlideShowApp"
	Solitaire - "Solitaire"
	System Info - "SystemInfo"
	System Setting - "SystemSetting"
	Tudou - "Tudou"
	Twitter - "Twitter"
	Youtube - "Youtube"
	Return Value:

	N/A
Note	LaunchServer's program must exist in the menu which has already removed the unused items. Otherwise, it could not be launched. For instance, if Youtube has already been removed from <i>menu.xml</i>, <code>launchService("Youtube")</code> could not be launched.

4.9 simpleGMI.play()

Function	<code>simpleGMI.play(url, mode)</code>
Description	Play the audio/video as specified in url
Parameter and Return Value	Parameter: url - specify the url address of the audio/video to be played. Use “;” to separate multiple files mode - play mode. 0 - Play once; 1 - Repeat Return Value: N/A
Note	N/A

4.10 simpleGMI.stopPlay()

Function	<code>simpleGMI.stopPlay()</code>
Description	Stop playing
Parameter and Return Value	Parameter: N/A Return Value: N/A
Note	N/A

4.11 simpleGMI.udp()

Function	<code>simpleGMI.udp(host, port, data)</code>
-----------------	--

Description	Send data to specific port of specific host
Parameter and Return Value	Parameter: host - destination host or IP address port - destination udp port data - send data contents (format: string) Return Value: N/A
Note	N/A

4.12 simpleGMI.post()

Function	simpleGMI. post(url, data, cb_post)
Description	Send http “POST” request to specific host
Parameter and Return Value	Parameter: url - Destination url data - Parameters sent with POST cb_post - The return function for the response to the POST request For example: function cb_post(data) The data parameter is for the response to the POST request. (If the data in the response is in xml format, GMI interface will transform it to JSON format. The data in other format in the response will remain the same). Return Value: N/A
Note	N/A

4.13 simpleGMI.exit()

Function	simpleGMI.exit()
Description	Exit from GMI. The application programs from GMI will be ended and the resource will be released.
Parameter and Return Value	Parameter:

	N/A Return Value: N/A
Note	N/A

4.14 simpleGMI.fullScreen()

Function	simpleGMI.fullScreen()
Description	Display in full screen. The control bar on the right hand side will not be displayed.
Parameter and Return Value	Parameter: N/A Return Value: N/A
Note	N/A

4.15 simpleGMI.normalScreen()

Function	simpleGMI.normalScreen()
Description	Normal display mode. The control bar on the right hand side will be displayed.
Parameter and Return Value	Parameter: N/A Return Value: N/A
Note	It is the default screen display mode.

5 GMIEngine Interface

5.1 GMIEngine.getNetworkInfo()

Function	GMIEngine.getNetworkInfo()
-----------------	----------------------------

Description	Obtain network information
Parameter and Return Value	Parameter: N/A Return Value: Return strings in JSON format. It has to be transformed to JSON format first. For instance: <pre>{ "mac": "00-0b-82-27-ea-ed", "ip": "192.168.1.118", "mask": "255.255.255.0", "type": "DHCP", "gateway": "192.168.1.1", "dns": "192.168.1.253", "nat": "Port Restricted Cone NAT (STUN)" }</pre> Strings in red will be displayed as the actual value in the phone.
Note	N/A

5.2 GMIEngine.getCurrentLanguage()

Function	GMIEngine.getCurrentLanguage()
Description	Obtain the current language
Parameter and Return Value	Parameter: N/A Return value: Return string in JSON format. It has to be transformed to JSON format first. For instance: <pre>{ "lan": "en" }</pre> Strings in red will be displayed as the actual value in the phone.
Note	N/A

5.3 GMIEngine.getCountry()

Function	GMIEngine.getCountry()
Description	Obtain the country code
Parameter and Return Value	Parameter:

	N/A Return Value: Return string in JSON format. It has to be transformed to JSON format first. For instance: <pre>{"country": "CN"}</pre> Strings in red will be displayed as the actual value in the phone.
Note	N/A

5.4 GMIEngine.getAccountInfo()

Function	GMIEngine. getAccountInfo()
Description	Obtain the account info
Parameter and Return Value	Parameter: N/A Return Value: Return string in JSON format. It has to be transformed to JSON format first. For instance: <pre>{ "info": [{ "index": "0", "enable": "1", "registerFlag": "1", "acctName": "IPVideoTalk", "server": "sip.ipvideotalk.com:48879", "userID": "8109060", "authID": "8109060", "callIDName": "8109060" }, { "index": "1", "enable": "1", "registerFlag": "1", "acctName": "8089", "server": "192.168.1.20", "userID": "8089", "authID": "8089", "callIDName": "8089" }, { "index": "2", "enable": "0", "registerFlag": "0", "acctName": "8011", "server": "192.168.1.20", "userID": "8011", "authID": "8011", "callIDName": "8011" }] }</pre> Strings in red will be displayed as the actual value in the phone.
Note	N/A

5.5 GMIEngine.put()

Function	GMIEngine. put(family, valuelist)
Description	Store the data specified by the parameter
Parameter and Return Value	Parameter: family – The family where the data belongs to valuelist – The data list to be set. The values should be separated by “&”. Return Value: N/A Example: The variables and function are as below: <pre>var ip = "192.168.1.220"; var name = "admin"; var password = "admin"; GMIEngine.put("HoneyWell", "ip=" + ip + "&name=" + name + "&password=" + password);</pre> Then ip/name/password will be stored and set under “HoneyWell” family.
Note	This interface can be used with get() so the application data can be read and written in GMI.

5.6 GMIEngine.get()

Function	GMIEngine.get(family, keylist)
Description	Get the data specified by the parameter
Parameter and Return Value	Parameter: family - The family where the data belongs to keylist – The data list to be retrieved. The values should be separated by “&”. Return Value: Return data in JSON format. The format should be transformed into JSON format first. Example:

	<pre>var valueList = GMIEngine.get("HoneyWell", "name&ip&password");</pre> If the variables have been set as below: <pre>var ip = "192.168.1.220"; var name = "admin"; var password = "admin";</pre> <pre>GMIEngine.put("HoneyWell", "ip=" + ip + "&name=" + name + "&password=" + password);</pre> Then the following expression <pre>var valueList = GMIEngine.get("HoneyWell", "name&ip&password")</pre> will return {"name": "admin", "ip": "192.168.1.220", "password": "admin"}
Note	This interface can be used with put() so the application data can be read and written in GMI.

6 GMIEngine Environment Variables

GMI provides global variables so that users could obtain the device information from them. The variable list and the descriptions are as below.

Variable Name	Description
GMIEngine.version	GMIEngine version
GMIEngine.ip	Phone's network IP address
GMIEngine.mask	Phone's subnet mask
GMIEngine.gateway	Phone's gateway IP
GMIEngine.dns	Phone's DNS IP
GMIEngine.mac	Phone's MAC address
GMIEngine.adressType	Three types to get the IP address: DHCP/static/PPPoE
GMIEngine.natType	NAT type
GMIEngine.accountActive	Account active or not.

	Array: GMIEngine.accountActive[0] GMIEngine.accountActive[1] GMIEngine.accountActive[2] are corresponding to the status of the three accounts
GMIEngine.accountName	Account name. Array: GMIEngine. accountName[0] GMIEngine. accountName[1] GMIEngine. accountName[2] are corresponding to the name of the three accounts
GMIEngine.accountServer	SIP server of the account. Array: GMIEngine. accountServer[0] GMIEngine. accountServer[1] GMIEngine. accountServer[2] are corresponding to the sip server of the three accounts
GMIEngine.accountUserID	UserID of the account Array: GMIEngine. accountUserID[0] GMIEngine. accountUserID[1] GMIEngine. accountUserID[2] are corresponding to the userID of the three accounts
GMIEngine.accountAuthID	AuthID of the account Array: GMIEngine. accountAuthID[0] GMIEngine. accountAuthID[1] GMIEngine. accountAuthID[2] are corresponding to the AuthID of the three accounts
GMIEngine.accountCallerID	CallerID of the account Array: GMIEngine. accountCallerID[0] GMIEngine. accountCallerID[1]

	GMIEngine. accountCallerID[2] are corresponding to CallerID of the three accounts
--	---

Users could use these variables directly in Javascript, for instance:

```
var account1Uri = GMIEngine.accountUserID[0] + "@" + GMIEngine.accountServer[0];  
var account2Uri = GMIEngine.accountUserID[1] + "@" + GMIEngine.accountServer[1];  
var account3Uri = GMIEngine.accountUserID[2] + "@" + GMIEngine.accountServer[2];
```