Logitech|G Arx Control Software Development Kit

Overview and Reference

Contents

Overview 3
SDK Package 3
Requirements 3
Interfacing with the SDK 3
Multiple clients using the SDK at the same time 3
Multiple mobile devices connected to Logitech Gaming Software 3
Multithread access to the SDK 3
Do's and Don'ts 4
Sample usage of the SDK 4
Using header and lib 4
How to use the context parameter in the initialization 4
Reference 6
ConfigOption Functions 6
LogiArxInit 8
LogiArxInitWithIcon 9
LogiArxAddFileAs 10
LogiArxAddContentAs 11
LogiArxAddUTF8StringAs 11
LogiArxAddImageFromBitmap 12
LogiArxSetIndex 12
LogiArxSetTagPropertyById 13
LogiArxSetTagsPropertyByClass 13
LogiArxSetTagContentById 14
LogiArxSetTagsContentByClass 14
LogiArxGetLastError 15
LogiArxShutdown 15
End-User License Agreement for Logitech|G Arx Control SDK 15

Overview

The Logitech|G Arx Control Software Development Kit enables applications such as games to interact with the Arx Control app on mobile devices.
Arx Control allows games and third party developers to take advantage of an iOS/Android device as a secondary screen to display useful data from the game.

SDK Package

The following files are included:

- LogitechGArxControlLib.h: C/C++ header file containing function prototypes
- LogitechGArxControlLib.lib: companion lib file to access DLL exported functions (32 and 64 bit)

Requirements

The Logitech|G Arx Control SDK can be used on the following platforms:

- Windows Vista (32-bit and 64-bit)
- Windows 7 (32-bit and 64-bit)
- Windows 8 (32-bit and 64-bit)

Logitech Gaming Software (8.52 or higher) needs to be running in order to use this SDK.

The SDK is a Windows based API for C/C++ programmers. The applet layout is based on web programming, so a basic knowledge of HTML/JavaScript/CSS is also needed.

Interfacing with the SDK

Using LogitechGArxControlLib.h and LogitechGArxControLibl.lib to access LogitechGArxControl.dll

The application can include LogitechGArxControlLib.h and link to LogitechGArxControlLib.lib (see "Sample usage of the SDK" further below or sample program in Samples folder). Installation folder for the DLL is under Logitech Gaming Software 8.55+ installation path, and it's automatically loaded by the static library LogitechGArxControlLib.lib.

Multiple clients using the SDK at the same time

The SDK allows more than one client to create an applet on Arx Control at the time. Logitech Gaming Software will handle and cache any request from the connected clients and as soon as a mobile device connect, each applet will be sent. The applet currently displayed in the foreground of Arx Control will have higher priority in content updates; all the other applets will still run in the background and receive content updates with a slower pace.

Multiple mobile devices connected to Logitech Gaming Software

Logitech Gaming Software will handle multiple connections with mobile devices. The applet will be sent to any connected device and the status will always be updated on any device.

Multithread access to the SDK

The SDK has been designed to be multithread safe, however, when using a multithreaded access to the Arx Control SDK, some guidelines need to be followed.

- The initialization and the shutdown of the SDK needs to happen on the same thread.
- Before shutting down, make sure that any other thread that is dealing with the SDK has completed the execution and there is no SDK function pending.

Do's and Don'ts

These are a few guidelines that may help you implement 'better' support in your game:

DO's

- Create a callback function and use it to be notified on applet focus change, device orientation change and user clicks on any object in your applet.
- The applet will be displayed on Arx Control only after the first valid call to LogiArxSetIndex. Send all your data first, and then set the index page to display correctly the app.

DON'TS

- Data is sent over Wi-Fi. Don't send oversized images or files. Try to optimize the format of any file that will be part of the applet.

Sample usage of the SDK

Using header and lib

Make sure to add the path to LogitechGArxControlLib.lib in your project settings. Usually this settings are Linker->Additional Library Directory and Linker->Input->Additional Dependencies. Also make sure that the path to the header LogitechGArxControlLib.h is included in your Additional Include Directory setting.

```
#pragma comment(lib, "LogitechGArxControlLib.lib")
#include "LogitechGArxControlLib.h"
```

How to use the context parameter in the initialization

The SDK communicates with Logitech Gaming Software in a separate thread in order to not slow down or block in any way the main thread of the game. On the other hand this means that the callback function will be executed from a separate thread. To be notified back on the main thread use the void pointer context parameter in the callback struct logiArxCbContext in LogiArxInit. Here is one example on how to take advantage of this capability. To look at a full compiling code, please refer to the sample DirectX_Sample in the package.

```
//Define a custom windows message
#define WM_ARXAPP_SDK_CALLBACK WM_APP + 1

static void __cdecl onCallback(int eventType, int eventValue, wchar_t * eventArg,
void *context)
{
	HWND main_hwnd = (HWND)context;
	//This struct is used in the PostMessage function to forward the callback
parameters.
	arxAppCallbackMessage *parameter = new arxAppCallbackMessage;
	parameter->eventType = eventType;
	parameter->eventValue = eventValue;
	wcsncpy_s(parameter->eventArg, eventArg, _TRUNCATE);
	//Using PostMessage to notify main thread of callback event
	PostMessage(main_hwnd, WM_ARXAPP_SDK_CALLBACK, reinterpret_cast<WPARAM>
(parameter) , static_cast<LPARAM>(0));
}
```

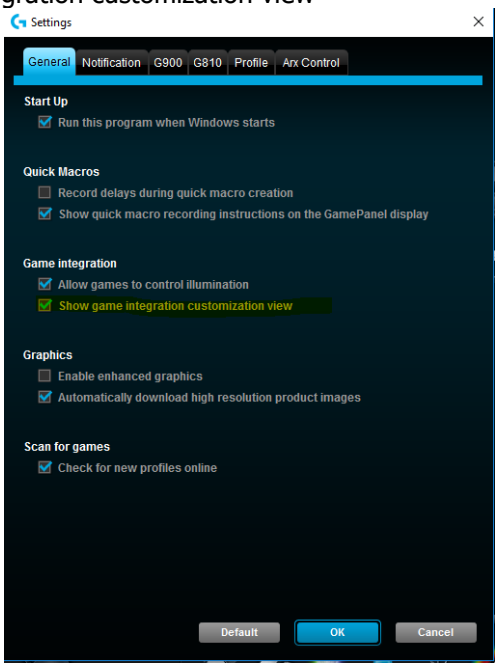
```
//Add this to the initialization code…
logiArxCbContext arxContextStruct;
arxContextStruct.arxCallBack = (logiArxCb)onCallback;

//Using the main window handle to be notified on the main thread through PostMessage
on the main handle
arxContextStruct.arxContext = m_hwnd;

if(!LogiArxInit(L"sdk.test.sample",L"My first ARX Applet", NULL))
{
	printf("Could not init ARX SDK. Error : %d\n",LogiArxGetLastError());
}


//In the windows message handling loop add this
case WM_ARXAPP_SDK_CALLBACK:
	{
		arxAppCallbackMessage *callbackMessageStruct =
				reinterpret_cast<arxAppCallbackMessage*> (wParam);
		printf("Received callback on main thread for ");
		switch(callbackMessageStruct->eventType)
		{
			case  LOGI_ARX_EVENT_FOCUS_INACTIVE:
			{
				printf("applet focus inactive \n");
			}
			break;

			case  LOGI_ARX_EVENT_FOCUS_ACTIVE:
			{
				if(callbackMessageStruct->eventValue ==
							LOGI_ARX_ORIENTATION_PORTRAIT)
				{
					printf("applet focus active in portrait mode\n");
				}
				if(callbackMessageStruct->eventValue ==
							LOGI_ARX_ORIENTATION_LANDSCAPE)
				{
					printf("applet focus active in landscape mode\n");

				}
			break;
			}

			case  LOGI_ARX_EVENT_MOBILEDEVICE_ARRIVAL:
			{
				printf("mobile device arrival\n");
				break;
			}

			case  LOGI_ARX_EVENT_MOBILEDEVICE_REMOVAL:
			{
				printf("mobile device removal\n");
```

```
                    break;
                    }

                    case  LOGI_ARX_EVENT_TAP_ON_TAG:
                    {
                           wprintf(L"click on tag with id %ls\n",
                                  callbackMessageStruct->eventArg);
                    break;
                    }

                    default:
                    {
                           printf("unknown message %d:%d",
                                  callbackMessageStruct->eventType,
                                  callbackMessageStruct->eventValue);
                    }
                    break;
}
```

Reference

ConfigOption Functions

The **LogiArxGetConfigOption** function set, allows the developer to query for an option set by the user and use that value to customize the interaction with the SDK. A call to any of these functions will create an entry in the Logitech Gaming Software – Applet Manager View. This view is disabled by default, since it's something targeting only "Advanced users", to enable it click on the Settings Icon in LGS and then check the box "Show Game integration customization view"




```
bool LogiArxGetConfigOptionNumber(wchar_t *configPath, double *defaultValue);

bool LogiArxGetConfigOptionBool(wchar_t *configPath, bool *defaultValue);

bool LogiArxGetConfigOptionColor(wchar_t *configPath, int *defaultRed, int
*defaultGreen, int *defaultBlue);

bool LogiArxGetConfigOptionString(wchar_t *configPath, wchar_t *defaultValue, int
bufferSize);

bool LogiArxSetConfigOptionLabel(wchar_t *configPath, wchar_t *label);
```

Parameters

- **configPath**: This identifies the option uniquely. This can be just a string (E.G. “Terrorist”) or it can be a two level tree (“Colors/Terrorist”). If the two level tree is specified, the option will be displayed in Logitech Gaming Software as an entry (“Terrorist”) inside a group (“Colors”).
- **defaultValue**: This parameter, depending on the specific function takes the default value for the relative option. If the option has been modified through LGS by the user, it will be filled in with

the modified value, otherwise the default value will be saved (to be shown to the user) and it won't be modified.

Return value
The function always returns true, unless some bad parameter has been specified.

Usage Example

```
bool arxMapEnabled = true;
LogiLedGetConfigOptionBool(L"map/enabled", &arxMapEnabled);
//This value will now contain the option as set by the user, or the default value if
//it has never been set.
if(arxMapEnabled)
{
	LogiArxAddFileAs(”map.png”, ”map.png“);
}
```

LogiArxInit
The **LogiArxInit** function makes sure there isn't already another instance running and then makes necessary initializations.

```
bool LogiArxInit(wchar_t * identifier, wchar_t * friendlyName, logiArxCbContext
callbackContest);
```

Parameters
- **identifier**: A unique identifier for the game or application. Text length is capped to 128 characters, any string longer than that will be truncated (e.g. com. company. game)
- **friendlyName**: the name that will be displayed on Arx Control for this applet. Text length is capped to 128 characters, any string longer than that will be truncated
- **callbackContest** : a logiArxCbContext struct containing the callback function reference and the context where to call it. See sample above to see how to use this. The context field in this struct can be NULL if there is no need to be notified in a different thread. The parameter can be NULL in case there is no need to be notified at all.

Return value
If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code. The return value is only dependent on the communication with Logitech Gaming Software. A true return value doesn't mean that there is a device and therefore a Arx Control running at the moment.
If the function fails and the error code is 6, this means that Logitech Gaming Software is not running at the moment, or the version running doesn't meet the minimum requirements for Arx Control. If this is the case do not send any file or update, everything will be lost. A suggested implementation is to check in a later moment either with some user interaction (from the game GUI) or with a timer.

Remarks
The callback function is the only way to be notified of a status change of the applet and therefore the only way to know that the applet is running.

The callback function is defined as follows:

```
typedef void (__cdecl *logiArxCb)(unsigned __int32 eventType, unsigned __int32
eventValue, wchar_t *eventArg, void * context);
```

This function will be called when an event occurs. Here are the possible values for the parameter *eventType*:

- LOGI_ARX_EVENT_MOBILEDEVICE_ARRIVAL. A mobile device is now connected.
  The parameter *eventArg* will be an empty string. The parameter *eventValue* will represent the type of device connected. Possible values are :
  - LOGI_ARX_DEVICETYPE_IPHONE
  - LOGI_ARX_DEVICETYPE_IPAD
  - LOGI_ARX_DEVICETYPE_ANDROID_SMALL
  - LOGI_ARX_DEVICETYPE_ANDROID_NORMAL
  - LOGI_ARX_DEVICETYPE_ANDROID_LARGE
  - LOGI_ARX_DEVICETYPE_ANDROID_XLARGE
  - LOGI_ARX_DEVICETYPE_ANDROID_OTHER
- LOGI_ARX_EVENT_MOBILEDEVICE_REMOVAL. No more devices connected to Logitech Gaming Software.
  The paramenter *eventArg* will be an empty string. The parameter *eventValue* will be 0.
- LOGI_ARX_EVENT_FOCUS_ACTIVE. The applet has received focus and is now in active status.
  The parameter *eventArg* will be an empty string. The parameter *eventValue* will represent the orientation in which the applet is being displayed. Possible values are
  - LOGI_ARX_ORIENTATION_PORTRAIT
  - LOGI_ARX_ORIENTATION_LANDSCAPE
- LOGI_ARX_EVENT_FOCUS_INACTIVE. The applet is now in background.
  The paramenter *eventArg* will be an empty string. The parameter *eventValue* will be 0.
- LOGI_ARX_EVENT_TAP_ON_TAG. The user has tapped on an element in the applet HTML active page.
  The parameter *eventArg* will represent the id of the HTML tag that has received a tap from the user. eventArg max length is 128 character, any tag id longer than that will be truncated. The parameter *eventValue* will be 0.

LogiArxInitWithIcon

The **LogiArxInit** function makes sure there isn't already another instance running and then makes necessary initializations. Use this function when you want to specify a custom icon for your applet instead of the exe default icon.

```
bool LogiArxInitWithIcon(wchar_t * identifier, wchar_t * friendlyName,
logiArxCbContext callbackContest, unsigned char *iconByteArray);
```

Parameters

- **identifier**: A unique identifier for the game or application. Text length is capped to 128 characters, any string longer than that will be truncated (e.g. com. company. game)

- **friendlyName**: the name that will be displayed on Arx Control for this applet. Text length is capped to 128 characters, any string longer than that will be truncated
- **callbackContest** : a logiArxCbContext struct containing the callback function reference and the context where to call it. See sample above to see how to use this. The context field in this struct can be NULL if there is no need to be notified in a different thread. The parameter can be NULL in case there is no need to be notified at all.
- **iconByteArray:** This parameter specifies the icon that will be shown for this applet on Arx Control applet bar. The icon is expected to be 144 x 144 in size and 4 bytes per pixel (R,G,B,A). The size of the byte array MUST be = LOGI_CUSTOMICON_BYTEARRAY_SIZE. For more info about the byte array ordering see the documentation for LogiArxAddImageFromBitmap. If this parameter is not specified, the applet will be shown on Arx Control using the icon of the executable that is interacting with the ARX SDK.

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code. The return value is only dependent on the communication with Logitech Gaming Software. A true return value doesn't mean that there is a device and therefore a Arx Control running at the moment.
If the function fails and the error code is 6, this means that Logitech Gaming Software is not running at the moment, or the version running doesn't meet the minimum requirements for Arx Control. If this is the case do not send any file or update, everything will be lost. A suggested implementation is to check in a later moment either with some user interaction (from the game GUI) or with a timer.

LogiArxAddFileAs

The **LogiArxAddFileAs** function sends a file from a local path to Arx Control.

```
bool LogiArxAddFileAs(wchar_t * filePath, wchar_t * fileName, wchar_t * mimeType = L"")
```

Parameters

- **filePath**: A string that represents a local path. This can be both relative to the game executable or absolute. Text length is capped to 256 characters, any string longer than that will be truncated
- **fileName**: A string that represents how the file will be referenced once loaded in Arx Control. Text length is capped to 256 characters, any string longer than that will be truncated
- **mimeType –** *Optional* - : A string that represents the mime type to associate the file with. This will determine how the app will interpret this file in the webview. This parameter is optional, the default value is an empty string. If the parameter is left empty the app will try to assign the right mime type to the file depending on its extension.

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks

The return value is only dependent on the communication with Logitech Gaming Software. A true return value doesn't mean that the file already made it all the way to the App.

LogiArxAddContentAs

The **LogiArxAddContentAs** function sends a block of memory to Arx Control.

```
bool LogiArxAddContentAs(const void* content, int size, wchar_t * fileName, wchar_t *
mimeType = L"")
```

Parameters

- **content**: A pointer to the block of memory to be sent to the Arx Control
- **size**: the size of the block of memory
- **filename** : A string that represents how the file will be referenced once loaded in Arx Control. Text length is capped to 256 characters, any string longer than that will be truncated
- **mimeType –** *Optional* - : A string that represents the mime type to associate the file with. This will determine how the app will interpret this file in the webview. This parameter is optional, the default value is an empty string. If the parameter is left empty the app will try to assign the right mime type to the file depending on its extension.

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks

If the size specified is bigger than the memory allocated for the pointer *content*, the function may raise an exception. If the size is smaller, only the first *size* byte will be sent.

LogiArxAddUTF8StringAs

The **LogiArxAddUTF8StringAs** function saves a string encoded in UTF8 to a file and sends it to Arx Control.

```
bool LogiArxAddUTF8StringAs(wchar_t* stringContent, wchar_t * fileName, wchar_t *
mimeType = L"")
```

Parameters

- **stringContent**: the string to be saved to file and sent over to Arx Control
- **filename**: the name of the file where the string will be saved and how it will be referenced once loaded in Arx Control. Text length is capped to 256 characters, any string longer than that will be truncated
- **mimeType –** *Optional* - : A string that represents the mime type to associate the file with. This will determine how the app will interpret this file in the webview. This parameter is optional, the default value is an empty string. If the parameter is left empty the app will try to assign the right mime type to the file depending on its extension.

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

LogiArxAddImageFromBitmap

The **LogiArxAddImageFromBitmap** function compresses the image in a png format and sends it to Arx Control.

```
bool LogiArxAddImageFromBitmap(BYTE bitmap[], int width, int height, wchar_t *
fileName);
```

Parameters

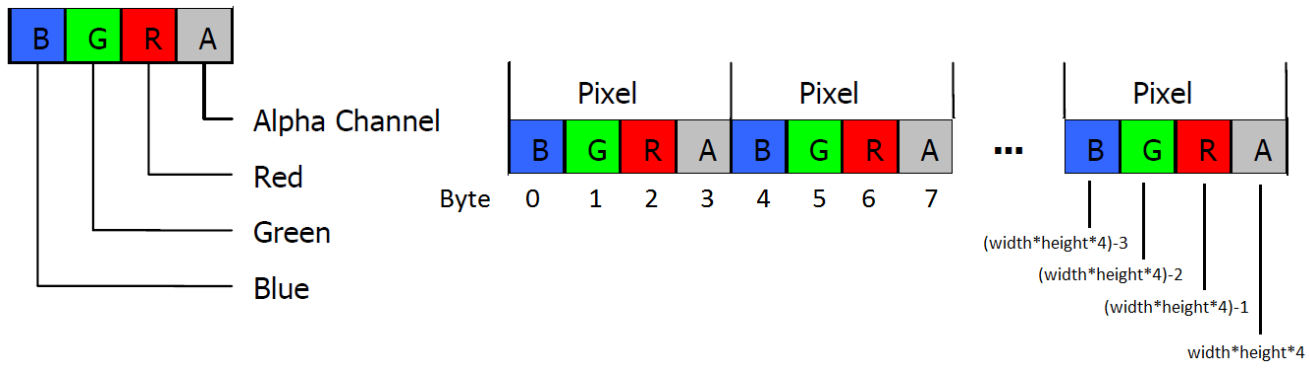
- **bitmap**: A BYTE array that identifies the image
- **width**: the width in pixel of the image
- **height**: the height in pixel of the image
- **filename:** A string that represents how the image will be referenced once loaded in Arx Control. The function will not automatically include the extension ".png". Text length is capped to 256 characters, any string longer than that will be truncated

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks

The illustration below shows the data arrangement required for the BYTE array bitmap.



Each of the bytes in the RGBA quad specify the intensity of the given value. The value ranges from 0 (the darkest color value) to 255 (brightest color value).
The expected allocated memory for the *bitmap* parameter is width * height * 4. If the memory allocated is less than what the function expects it may cause an exception.
The mime type associated with this file will always be "image/png", since the file is automatically compressed by Logitech Gaming Software in the PNG format.

LogiArxSetIndex

The **LogiArxSetIndex** function sets which page is the one to be displayed on Arx Control.

```
bool LogiArxSetIndex(wchar_t * fileName);
```

Parameters

- **fileName**: A string that represents the file to display on Arx Control. This can be a file that has been sent with any of the above function calls. Text length is capped to 256 characters, any string longer than that will be truncated

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks

The first time this function is called on a valid file the applet will be brought in the foreground on Arx Control.

LogiArxSetTagPropertyById

The **LogiArxSetTagPropertyById** function updates a tag property in the applet html pages.

```
bool LogiArxSetTagPropertyById(wchar_t * tagId, wchar_t * prop, wchar_t *newValue);
```

Parameters

- **tagId**: The id of the tag to update the property on. Text length is capped to 128 characters, any string longer than that will be truncated
- **prop**: The property to update. Text length is capped to 128 characters, any string longer than that will be truncated
- **newValue:** The new value to assign to the property *prop* on the tag with id *tagId.* Text length is capped to 128 characters, any string longer than that will be truncated

Return value

If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks

If the no tag with id tagId is found, the function will return true and no tag will be updated. If more than one tag on any page have the same id, each one of them will be updated.

LogiArxSetTagsPropertyByClass

The **LogiArxSetTagsPropertyByClass** function updates a property on a class of tags in the applet html pages.

```
bool LogiArxSetTagsPropertyByClass(wchar_t * tagsClass, wchar_t * prop, wchar_t
*newValue);
```

Parameters

- **tagClass**: The class of the tags to update the property on. Text length is capped to 128 characters, any string longer than that will be truncated

- **prop**: The property to update. Text length is capped to 128 characters, any string longer than that will be truncated
- **newValue:** The new value to assign to the property *prop* on all the tags with class *tagClass.* Text length is capped to 128 characters, any string longer than that will be truncated

Return value
If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks
If the no tag with class tagClass is found, the function will return true and no tag will be updated.

LogiArxSetTagContentById

The **LogiArxSetTagContentById** function updates the content of a tag in the applet html pages.

```
bool LogiArxSetTagContentById(wchar_t * tagId, const wchar_t *newContent);
```

Parameters
- **tagId**: The id of the tag to update the property on. Text length is capped to 128 characters, any string longer than that will be truncated
- **newContent:** The new html content that will be injected in the tag.

Return value
If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks
If the no tag with id tagId is found, the function will return true and no tag content will be updated.

LogiArxSetTagsContentByClass

The **LogiArxSetTagsContentByClass** function updates the content of a class of tags in the applet html pages.

```
bool LogiArxSetTagsContentByClass(wchar_t * tagsClass, const wchar_t *newContent)
```

Parameters
- **tagClass**: The class of the tags for which the content will be replaced. Text length is capped to 128 characters, any string longer than that will be truncated
- **newContent:** The new html content that will be injected in the tags.

Return value
If the function succeeds, it returns true. Otherwise false. Call LogiArxGetLastError to get the error code

Remarks
If the no tag with class tagClass is found, the function will return true and no tag will be updated.

LogiArxGetLastError

The **LogiArxGetLastError** function retrieves and returns the last error occurred in the SDK function calls. Call this function if an SDK function call fails to get more detailed information on the failure.

```
int LogiArxGetLastError();
```

Return value

The return value indicates the error code according to this table:

ERROR CODE	ERROR
0	Success
1	Wrong parameter format
2	Null parameter not supported
3	Wrong file path
4	SDK not initialized
5	SDK already initialized
6	Connection with Logitech Gaming Software broken
7	Error creating thread
8	Error copying memory

Remarks

When any of the function calls returns error code 6 – Connection with Logitech Gaming Software broken, is a sign that Logitech Gaming Software will not receive any of the subsequent calls. The applet will be removed from the active on Arx Control. The suggested procedure in this case is to call LogiArxShutdown to prevent any memory leak.

LogiArxShutdown

The **LogiArxShutdown** function frees the memory used by the SDK for the applet and shuts it down on Arx Control.

```
void LogiArxShutdown();
```

Remarks

If using a multithreaded environment, call this function only any other thread that was dealing with the Arx Control SDK has finished its execution. Calling LogiArxShutdown while other function calls are pending might cause unexpected behavior.

End-User License Agreement for Logitech|G Arx Control SDK

This End-User License Agreement for Logitech Gaming ARX CONTROL SDK (“Agreement”) is a legal agreement between you, either an individual or legal entity (“You” or “you”) and Logitech Inc. (“Logitech”) for use of

the Logitech Gaming ARX CONTROL software development kit, which includes computer software and related media and documentation (hereinafter “Logitech Gaming ARX CONTROL SDK”). By using this Logitech Gaming ARX CONTROL SDK, you are agreeing to be bound by the terms and conditions of this Agreement. If you do not agree to the terms and conditions of this Agreement, promptly return the Logitech Gaming ARX CONTROL SDK and other items that are part of this product in their original package, or if you have downloaded this software from a Logitech or a Distributor web site, then you must stop using the software and destroy any copies of the software in your possession or control.

1 Grant of License and Restrictions. This Agreement grants You the following rights provided that You comply with all terms and conditions of this Agreement.

(a) Logitech grants You a limited, non-exclusive, nontransferable license to install and use an unlimited number of copies of the Logitech Gaming ARX CONTROL SDK on computers. All other rights are reserved to Logitech.

(b) You shall not reverse engineer, decompile or disassemble any portion of the Logitech Gaming ARX CONTROL SDK, except and only to the extent that this limitation is expressly prohibited by applicable law.

(c) At your option, you may provide reasonable feedback to Logitech, including but not limited to usability, bug reports and test results, with respect to the Logitech Gaming ARX CONTROL SDK. All bug reports, test results and other feedback provided to Logitech by You shall be the property of Logitech and may be used by Logitech for any purpose.

(d) In the event Logitech, in its sole discretion, elects to provide copies of the Logitech Gaming ARX CONTROL SDK to more than one individual employed by You (if You are not a single individual), each such individual shall be entitled to exercise the rights granted in this Agreement and shall be bound by the terms and conditions herein.

2 Updates. Logitech is not obligated to provide technical support or updates to You for the Logitech Gaming ARX CONTROL SDK provided to You pursuant to this Agreement. However, Logitech may, in its sole discretion, provide further pre-release versions, technical support, updates and/or supplements (“Updates”) to You, in which case such Updates shall be deemed to be included in the “Logitech Gaming ARX

CONTROL SDK" and shall be governed by this Agreement, unless other terms of use are provided in writing by Logitech with such Updates.

3 **Intellectual Property Rights.** The Logitech Gaming ARX CONTROL SDK is licensed, not sold, to You for use only under the terms and conditions of this Agreement. Logitech and its suppliers retain title to the Logitech Gaming ARX CONTROL SDK and all intellectual property rights therein. The Logitech Gaming ARX CONTROL SDK is protected by intellectual property laws and international treaties, including U.S. copyright law and international copyright treaties. All rights not expressly granted by Logitech are reserved.

4 **Disclaimer of Warranty.** TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, LOGITECH, ITS SUPPLIERS AND DISTRIBUTORS PROVIDE THE LOGITECH GAMING ARX CONTROL SDK AND OTHER LOGITECH PRODUCTS AND SERVICES (IF ANY) AS IS AND WITHOUT WARRANTY OF ANY KIND. LOGITECH AND ITS SUPPLIERS AND DISTRIBUTORS EXPRESSLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS WITH RESPECT TO THE LOGITECH GAMING ARX CONTROL SDK AND ANY WARRANTIES OF NON-INTERFERENCE OR ACCURACY OF INFORMATIONAL CONTENT. NO LOGITECH DISTRIBUTOR, AGENT, OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATION, EXTENSION, OR ADDITION TO THIS WARRANTY. Some jurisdictions do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.

5 **Limitation of Liability.** IN NO EVENT WILL LOGITECH, ITS SUPPLIERS, OR DISTRIBUTORS BE LIABLE FOR ANY COSTS OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR SERVICES, LOST PROFITS, LOSS OF INFORMATION OR DATA, OR ANY OTHER SPECIAL, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING IN ANY WAY OUT OF THE SALE OF, USE OF, OR INABILITY TO USE THE LOGITECH GAMING ARX CONTROL SDK OR ANY LOGITECH PRODUCT OR SERVICE, EVEN IF LOGITECH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO CASE SHALL LOGITECH'S, ITS SUPPLIERS' AND DISTRIBUTORS' TOTAL LIABILITY EXCEED THE ACTUAL MONEY PAID FOR THE LOGITECH PRODUCT OR SERVICE GIVING RISE TO THE LIABILITY. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential

damages, so the above limitation or exclusion may not apply to you. The above limitations will not apply in case of personal injury where and to the extent that applicable law requires such liability.

6 **U.S. Government Rights.** Use, duplication, or disclosure of the software contained in the Logitech Gaming ARX CONTROL SDK by the U.S. Government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988) FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Logitech Inc. 7600 Gateway Blvd, Newark, CA 94560.

7 **Export Law Assurances.** You agree and certify that neither the Logitech Gaming ARX CONTROL SDK nor any other technical data received from Logitech will be exported outside the United States except as authorized and as permitted by the laws and regulations of the United States. If you have rightfully obtained the Logitech Gaming ARX CONTROL SDK outside of the United States, you agree that you will not re-export the Logitech Gaming ARX CONTROL SDK nor any other technical data received from Logitech, except as permitted by the laws and regulations of the United States and the laws and regulations of the jurisdiction in which you obtained the Logitech Gaming ARX CONTROL SDK.

8 **Termination:** This Agreement is effective until terminated. Upon any violation of any of the provisions of this Agreement, or any provisions of any agreement between you and a Distributor, rights to use the Logitech Gaming ARX CONTROL SDK shall automatically terminate and the Logitech Gaming ARX CONTROL SDK must be returned to Logitech or all copies of the Logitech Gaming ARX CONTROL SDK destroyed. You may also terminate this Agreement at any time by destroying all copies of the Logitech Gaming ARX CONTROL SDK in your possession or control. If Logitech makes a request via public announcement or press release to stop using the copies of the Logitech Gaming ARX CONTROL SDK, you will comply immediately with this request. The provisions of paragraphs 3, 7, 8 and 12 will survive any termination of this Agreement.

9 **General Terms and Conditions.** If You are an individual signing this Agreement on behalf of a company, then You represent that You have authority to execute this Agreement on behalf of such company. This Agreement will be governed by and construed in accordance with the laws of the United States and the State of California, without regard to or application of its choice of law rules or principles. If for any reason a court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable,

that provision of the Agreement shall be enforced to the maximum extent permissible so as to affect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect. This Agreement constitutes the entire agreement between You and Logitech respect to the use of the Logitech Gaming ARX CONTROL SDK and supersedes all prior or contemporaneous understandings, communications or agreements, written or oral, regarding such subject matter.