



Logitech G-key Plugin Development Guidelines For Headsets

Rev.	Date	Comment
001	June 11, 2010	Initial developer release

Table of Contents

Introduction	3
Definitions	3
Required developer tools	4
G-key plugin overview	4
G-key programmable module library format	4
DLL preparation	4
Limitation for loading the user G-key DLL	4
G-key DLL interface functions	4
G-key plugin interface	5
Managing G-key plugins	6
Adding a G-key plugin	6
Removing a G-key plugin	7
Testing G-key plugins	7
Contact	8

Introduction

Select Logitech gaming headsets feature G-keys that users can customize to control background applications even while playing games at full-screen.

Headset G-keys are customized via Logitech gaming headset software included with G-key enabled headsets. The software comes with a handful of preset G-key actions for common applications including media players and chat clients.

Recent models of G-key enabled headsets feature the ability to expand G-key customization options through G-key plugins.

This document provides application developers and independent programmers instructions for how to create and test G-key plugins. Included within is sample code and a sample project template for developer to write their own G-key plugin.

Definitions

Term	Definition
API	Application program interface function
Developer	Person or company creating the G-key plugin
DLL	Windows Win32 dynamic linked library
G930	Logitech Gaming Headset G930; the first headset to feature G-key plugin capabilities
G-key	Customizable button on the headset; G-keys referenced in this document only pertain to G-keys on headsets
G-key plugin	Expands G-key customization options once installed in the Logitech gaming headset software
HW	Hardware (headset) device
Logitech gaming headset software	Software included with select gaming headsets that enable suser to customize the G-keys
Programmable module	A user developed application library for G-key
SW	Logitech Gaming headset software
UI	User interface elements in the software

Required developer tools

The Developer needs a Microsoft Visual Studio and VC++ compiler or equivalent development tool to produce a G-key plugin that is compatible with Windows OS version that currently running the headset application software.

Included in the SDK is a sample application project template called "GKeyDLLSample1", which has prepared with Microsoft Visual Studio 2005 and VC++ 8.0 compiler. The develop may use this sample project expand it for a specific needs.

The Developer must be able to understand C or C++ programming language in order to understand a Logitech provided sample project template along with the development guidelines so that the headset software can work with their plugin.

The Developer will also need a Logitech Gaming Headset G930 plugged into system with software installed in order to use the Logitech gaming headset software to test the G-key plugin.

G-key plugin overview

G-key programmable module library format

The library must be a dynamic linked library (DLL) that can be loaded and used with G-keys.

DLL preparation

DLL is required to be statically linked in all the related libraries and components that are used in the DLL excepts the component(s) that is available in Windows.

Limitation for loading the user G-key DLL

The gaming headset software only loads the first 100 user DLLs in addition to the Logitech predefined applications and 128 action commands per DLL. The user DLL (DLL file name) will be added to the end of the existing application list in the G-key UI.

G-key DLL interface functions

Each DLL is required to expose 2 APIs for the headset software to use the DLL. The functions are defined as:

WCHAR** **GetGkeyCommandList** (unsigned int LanguageCode)

BOOL **RunGkeyCommand** (unsigned int CommandID)

The function name is case sensitive character.

G-key plugin interface

WCHAR** GetGkeyCommandList (unsigned int LanguageCode).

Parameters

LanguageCode - A language code for the language that is currently used for the headset software.

The languages that support by Logitech G series headset software are:

No.	LanguageCode	Languages
1.	1028	Chinese (Traditional)
2.	1029	Czech
3.	1030	Danish
4.	1031	German
5.	1033/default	English
6.	1034	Spanish(Spain)
7.	1035	Finnish
8.	1036	French
9.	1038	Hungarian
10.	1040	Italian
11.	1042	Korean
12.	1043	Dutch
13.	1044	Norwegian
14.	1045	Polish
15.	1046	Portuguese(Brazil)
16.	1053	Swedish
17.	2070	Portuguese(Portugal)
18.	1049	Russian
19.	2052	Chinese (Simplified)

Return value

A list of Unicode character strings. Each string is an action command that is supported by the DLL. The software supports up to 128 command strings per DLL. If the function returns a NULL command list, the software will not load this DLL into the G-key UI. A command list must be terminated by a **NULL** command string so that the application knows where a command list is ended. Without a NULL terminated string, it can cause unpredictable problem to the application while it constructs the command list.

For example:

If a module has 2 action commands, such as "Sound Mute" and "Microphone Mute" then a return command list would be constructing as the following.

```
static const WCHAR* EnglishCmdList[] = { L"Sound Mute", L"Microphone Mute", NULL };
```

Remarks

This function returns a list of action commands for a specified languageCode that supported by the DLL. If the requesting language is not supported by the DLL, it can return a default language's command list or return a NULL command list to skip loading this DLL.

The headset software invokes this function to execute an action command for a given CommandID.

BOOL RunGkeyCommand (unsigned int CommandID)

Parameters

CommandID - Is a location of a command in the command list, the value is range from (0 to n-1). 0 represents a first command and n-1 is a last command in a command list.

Return value

TRUE if a commandID is a valid value and the function succeeds.

FALSE if a command ID is an unknown and no action has been taken.

Remarks - This function performs an action for a given command ID.

Importance note:

In order for the headset software to recognize these 2 functions, the function name must be the same name as defined above and case sensitive character.

Managing G-key plugins

Upon the launch of the Logitech gaming headset software, the G-key libraries are loaded into the UI and available to the user for G-key customization. Users can add and remove additional G-key plugins in the UI.

Adding a G-key plugin

To add a G-key plugin, click on a "+" button in the UI. A Windows explorer dialog is popup and enables the user to navigate to the G-key plugin DLL file location. Double clicking on the .DLL file or selecting the DLL File and choosing "Open" button will load the G-key plugin into the Logitech gaming headset software.

Upon loading the G-key plugin, the DLL will move into the product's user application designated directory (folder). The default path is dependent on the OS version.

On Windows XP it's likely to be at "%USER_DEFAULT_DRIVE%:\documents and settings\All Users\Application Data\Logitech\ G-Series Headset\GkeyAppls" if a new DLL name already exists in this folder, it get replace with a new file.

Removing a G-key plugin

To remove a G-key plugin, click on the "-" button in the UI. From the new dialog window, select one or more G-key plugins plug and then click "Remove". To close the dialog window, click the "X" in the upper right corner.

When the user removes a G-key plugin, the respective application and actions will be removed the the G-key dropdown menus in the main UI, and the DLL file will be archived in the **GkeyAppls** folder. Users should not edit or delete a "GkeyUserApps.rec" because this file contains the DLL(s) names those are currently used in with the headset software.

Testing G-key plugins

Compile the programmable module DLL under development. Upon successful compilation, add the G-key plugin through the Logitech gaming headset software UI, assign the respective application and action command to a G-key, and press the G-key button from the headset to verify the G-key function.

If the user's application module is not shown in the G-key application list, inspect the application loading result log file to see if there is any error has occurred while the headset software loading the DLLs. The error message is only in **English language** and output to a file "**LoadDLLResult.log**" in a "%USER_DEFAULT_DRIVE%:\Documents and Settings\All Users\Application Data\Logitech\G-Series Headset\Resultlogs" folder.

The error messages will be one of the following listing:

"Error: library x – cannot find API function GetGkeyCommandList()"

"Error: library x – cannot find API finction RunGkeyCommand() "

"Error: library x – contains no action command"

"Information: the G-key user library path is not set or does not exist"

Contact

For technical support pertaining to G-key plugin development for headsets, contact AudioGkeySWsupport@logitech.com.

If you'd like to get in contact with our gaming business development team, email bizrel@logitech.com.