



COMMON

INDEX

INTRODUCTION	3
COMMANDS	4
XPEAK_COMMAND_GET_CAPABILITIES (v.0706)	5
XPEAK_COMMAND_GET_COMMANDS (v.0706)	6
XPEAK_COMMAND_GET_CONFIGURATION (v.0706)	7
XPEAK_COMMAND_GET_SERIAL_NUMBER (v.2105)	8
XPEAK_COMMAND_GET_STATUS (v.0706)	9
XPEAK_COMMAND_INITIALIZE (v.0706)	10
XPEAK_COMMAND_RESET (v.0706)	11
XPEAK_COMMAND_SET_CONFIGURATION (v.0706)	12
XPEAK_COMMAND_SHUTDOWN (v.0706)	13
EVENTS	14
RESULTS	16

INTRODUCTION

This documentation details the of commands that are common for every Xpeak service. Sometimes, certain *common commands* can be overwritten within a specific service, because they change their behavior.

All commands described here meet the *Xpeak* specification [General Message Format](#)

COMMANDS

 **XPEAK_COMMAND_GET_CAPABILITIES** (0x7060017)

Version: 0706

Description:

This command is used to get the capabilities of the device. The result fields of this command will depend on the service type. Depending on their values, the application behavior should be different and should be adapted to the peripheral characteristics.

 **XPEAK_COMMAND_GET_COMMANDS** (0x7060064)

Version: 0706

Description:

Gets the list of commands that the service accepts, including the versions supported. This command can be executed before the [XPEAK_COMMAND_INITIALIZE](#) command.

 **Result:**

- **Struct[] Commands**

The list of commands. The fields of this struct are:

- **int Id**
Command code.
- **int FromVersion**
Lower version supported.
- **int ToVersion**
Upper version supported.

XPEAK_COMMAND_GET_CONFIGURATION (0x7060020)

Version: 0706

Description:

Gets the configuration of the service. This command can be executed before the [XPEAK_COMMAND_INITIALIZE](#) command.

The application can change the configuration of the service using the [XPEAK_COMMAND_SET_CONFIGURATION](#) command.

Result:

- **Struct[] Fields**

The list of fields with the configuration of the service. The fields contained on this struct are:

- **String Name**

Name of the configuration parameter.

- **int Type**

Type of field. Can be one of the following values:

- XPEAK_FIELD_TYPE_BOOLEAN (0x7060021)
- XPEAK_FIELD_TYPE_STRING (0x7060022)
- XPEAK_FIELD_TYPE_INTEGER (0x7060023)
- XPEAK_FIELD_TYPE_INTEGER_HEXADECIMAL (0x7060026)
- XPEAK_FIELD_TYPE_DOUBLE (0x7060024)
- XPEAK_FIELD_TYPE_CHARACTER (0x7060025)
- XPEAK_FIELD_TYPE_LONG (0x7060054)
- XPEAK_FIELD_TYPE_BASE64 (0x7060055)
- XPEAK_FIELD_TYPE_COLOR (0x7060056)

- **String Options**

If the field can have a set of predefined values, this field will contain a list of them, separated by commas. For instance, if the field is the parity of the serial communications, this field will contain the value: *EVEN, ODD, NONE*.

This field is optional.

- **boolean IsEditable**

Indicates if the field is editable. This parameter is only useful for graphical tools that allow to change the configuration parameters.

- **Base64 Value**

Value of the configuration field.

 **XPEAK_COMMAND_GET_SERIAL_NUMBER** (0x7060116)

Version: 2105

Description:

Returns the serial number of the device

 **Result:**

- **String id**
Contains the serial number of the device.

 **XPEAK_COMMAND_GET_STATUS** (0x7060051)

Version: 0706

Description:

This command is used to get the status of the device.

 **Result:**

- **int[] Statuses**

The possible values are:

Status	Description
XPEAK_STATUS_WORKING (0x706003C)	The device is working properly.
XPEAK_STATUS_HARDWARE_ERROR (0x7060040)	There is a hardware error in the device and it is necessary to solve it with a technical manipulation.
XPEAK_STATUS_USER_ACTION_ERROR (0x7060041)	The device has an error that can be solved by the user.
XPEAK_STATUS_CONNECTION_ERROR (0x7060042)	Error trying to communicate with the device.
XPEAK_STATUS_BUSY (0x706003F)	The device is being used by another application.

 **XPEAK_COMMAND_INITIALIZE** (0x7060015)

Version: 0706

Description:

Initializes the service, communicating, if needed, with the physical device. The only commands that can be sent to the service before this one are:

- [XPEAK_COMMAND_GET_COMMANDS](#)
- [XPEAK_COMMAND_SET_CONFIGURATION](#)
- [XPEAK_COMMAND_GET_CONFIGURATION](#)

Executing this command more than once will not return an error, it will just increase the counter of initializations done. The command [XPEAK_COMMAND_SHUTDOWN](#) will be correctly executed when it is invoked as many times as the initialize command.

The returned result will always be [XPEAK_RESULT_OK](#) (0x7060001), indicating that the connection between the application and the service is established. This does not mean that the device is in a proper state, so a call to the [XPEAK_COMMAND_GET_STATUS](#) command should be done to know the real device state.

 **XPEAK_COMMAND_RESET** (0x7060004)

Version: 0706

Description:

This command executes a reset of the device. In some devices, the execution of this command is required to recover it from some errors.

 **Result:**

See Common Results

XPEAK_COMMAND_SET_CONFIGURATION (0x7060016)

Version: 0706

Description:

This command allows to change the initial configuration of the service. This command can be called before the call of the [XPEAK_COMMAND_INITIALIZE](#) command.

The current configuration data can be obtained executing the [XPEAK_COMMAND_GET_CONFIGURATION](#) command.

Parameters:

- **Struct[] Fields**

The list of fields with the new configuration of the service. The fields of this struct are the following:

- **String Name**

Name of the configuration parameter

- **Base64 Value**

New value of the configuration parameter.

Result:

See Common Results

 **XPEAK_COMMAND_SHUTDOWN** (0x7060063)

Version: 0706

Description:

This command frees the resources used by the service and closes the connections between the service and the physical device. If more than one [XPEAK_COMMAND_INITIALIZE](#) command have been executed, then the service will only decrease the internal counter.

 **Parameters:**

- **boolean Force**
If the value is *true*, the service will do the shutdown independently of the internal counter of INITIALIZES executed.

 **Result:**

See Common Results

EVENTS

Description:

Some commands require to send events, besides the answer itself. For instance, in case of the ejection of a card in a card reader, the answer will indicate if the ejection has been executed correctly and, later, an event will be sent to indicate the application that the card has been taken by the user, or that an error has happened (timeout or card jam, etc).

Events are sent with the same identifier as the command that generated them. Besides this identifier, all the events share, as a minimum, these fields:

Type	Name	Description
int	Result	The result returned by the event. The values that can be returned are the ones documented in this page and the specific codes of each command.
String[]	EventDetails	In some events, and depending on the returned result, this field can be included with some extra information. These details can vary depending on the vendor. This field must only be considered during the development phase and to trace information or help the developer.

Besides these fields, each event can define as many fields as needed.

Following is the list of the common events, indicating for each one the Result field described above.

🔌 Events:

XPEAK_RESULT_TIMEOUT (0x7060002) Since 0706

The time to execute the command has expired.

XPEAK_RESULT_DISABLED (0x7060173) Since 0706

An "EnableXxxxx" command has been disabled using the corresponding "DisableXxxxx" command.

XPEAK_RESULT_HARDWARE_ERROR (0x7060059) Since 0706

Hardware error in the device.

XPEAK_RESULT_USER_ACTION_ERROR (0x706005A) Since 0706

There is an error in the device that can be solved by the user. For instance, the cover of the printer is open or there are notes to remove in the dispenser. Some services can use the field *EventDetails* to give more information.

XPEAK_RESULT_CONNECTION_ERROR (0x706005B) Since 0706

Error trying to communicate with the device. The field *EventDetails* can provide more information.

RESULTS

Description:

The basic structure of the answer messages (common to every command), contains, as a minimum, two fields:

Type	Name	Description
int	Result	The returned result after the command execution. The values that can be returned are the ones documented in this page and the specific codes of each command.
String[]	ResultDetails	In some commands, and depending on the returned result, this field can be included with some extra information. These details can vary depending on the vendor. This field must only be considered during the development phase and to trace information or help the developer.

Each command can define, besides these, as many fields as needed.

 **Result:**

- **int Result**

Besides the values specified in each command, every Xpeak command can return one of the following values:

Result	Description
XPEAK_RESULT_OK (0x7060001)	The command finished successfully
XPEAK_RESULT_COMMAND_NOT_SUPPORTED (0x706001D)	The service does not support the required command.
XPEAK_RESULT_VERSION_NOT_SUPPORTED (0x706001F)	The service supports the command, but not the version specified.
XPEAK_RESULT_INVALID_PARAMETER (0x706003A)	Some of the parameters is not correct. Some services will provide extra information about the invalid parameters in the <i>ResultDetails</i> field.
XPEAK_RESULT_NOT_INITIALIZED (0x7060057)	The service was not initialized.
XPEAK_RESULT_BUSY (0x7060058)	The command can not be executed because the device is busy whether by another command or because it is being used by another user.
XPEAK_RESULT_HARDWARE_ERROR (0x7060059)	There is a hardware error that makes not possible to execute the command.
XPEAK_RESULT_USER_ACTION_ERROR (0x706005A)	There is an error in the device that makes not possible to execute the command. This error can be solved by the user itself. For instance, the printer cover is open or there are notes to be taken in the dispenser. Some services can use the field <i>EventDetails</i> to provide more information.
XPEAK_RESULT_CONNECTION_ERROR (0x706005B)	Error trying to communicate with the device. The field <i>EventDetails</i> can provide more information
XPEAK_RESULT_SEQUENCE_ERROR (0x706009A)	Sequence error. The command can not be executed in this moment due to a sequence error. Perhaps it is necessary to execute another command before this one.
XPEAK_RESULT_SOFTWARE_ERROR (0x70600C2)	There is a software error not controlled (i.e. an exception).
XPEAK_RESULT_COMMAND_ERROR (0x7060143)	The command could not be executed or was not accepted by the device.

- **String[] ResultDetails**

In some commands, and depending on the returned result, this field can be included with some extra information. These details can vary depending on the vendor. This field must only be considered during the development phase and to trace information or help the developer.