Ambarella

A-Series Wireless Connectivity API:

Remote Control

Version 4.1.0

June 10, 2015

Confidentiality Notice:

US
3101 Jay Street
Ste. 110
Santa Clara, CA 95054, USA
Phone: +1.408.734.8888
Fax: +1.408.734.0788

Taiwan
Suite C1, No. 1, Li-Hsin Road 1
Science-Based Industrial Park
Hsinchu 30078, Taiwan
Phone: +886.3.666.8828
Fax: +886.3.666.1282

Korea
6 Floor, Hanwon-Bldg.
Sunae-Dong, 6-1, Bundang-Gu
SeongNam-City, Kyunggi-Do
Republic of Korea 463-825
Phone: +031.717.2780
Fax: +031.717.2782

Japan - Yokohama
Shin-Yokohama Business Center Bldg. 5th Floor
3-2-6 Shin-Yokohama, Kohoku-ku,
Yokohama, Kanagawa, 222-0033, Japan
Phone: +81.45.548.6150
Fax: +81.45.548.6151

Hong Kong
Unit A&B, 18/F, Spectrum Tower
53 Hung To Road
Kwun Tong, Kowloon
Phone: +85.2.2806.8711
Fax: +85.2.2806.8722

China - Shenzhen
Unit E, 5th Floor
No. 2 Finance Base
8 Ke Fa Road
Shenzhen, 518057, China
Phone: +86.755.3301.0366
Fax: +86.755.3301.0966

China - Shanghai
9th Floor, Park Center
1088 Fangdian Road, Pudong New District
Shanghai 201204, China
Phone: +86.21.6088.0608
Fax: +86.21.6088.0366

I Contents

II Preface

This document provides technical details using a set of consistent typographical conventions to help the user differentiate key concepts at a glance.

Conventions include:

Example	Description
AmbaGuiGen, **DirectUSB** **Save, File > Save** **Power, Reset, Home**	Software names GUI commands and command sequences Computer / Hardware buttons
Flash_IO_control **da**, **status**, **enable**	Register names and register fields. For example, **Flash_IO_control** is the register for global control of Flash I/O, and bit 17 (**da)** is used for DMA acknowledgement.
GPIO81, **CLK_AU**	Hardware external pins
VIL, VIH, VOL, VOH	Hardware pin parameters
INT_O, RXDATA_I	Hardware pin signals
amb_performance_t **amb_operating_mode_t** **amb_set_operating_mode()**	API details (e.g., functions, structures, and type definitions)
`/usr/local/bin` `success = amb_set_operating_mode (amb_base_address, & operating_mode)`	User entries into software dialogues and GUI windows File names and paths Command line scripting and Code

Table II-1. Typographical Conventions for Technical Documents.

Additional Ambarella typographical conventions include:

- Acronyms are given in UPPER CASE using the default font (e.g., AHB, ARM11 and DDRIO).
- Names of Ambarella documents and publicly available standards, specifications, and databooks appear in *italic* type.

1 Overview

1.1 Overview: Introduction

This document provides the Remote Control Application Programming Interface (API) for Ambarella wireless-connectivity digital camera chips. The API commands described in this document offer flexibility that enables a wide range of handheld devices (such as smartphones and tablets) to control and execute commands on one (or more) Ambarella-based camera products.

The API commands in this document can be used with Ambarella A-series chips (A5s and later) running a Wireless or Connected Camera SDK. The Remote Control API is accessible from the Application Layer of the Camera Software Architecture.

1.2 Overview: Background

This Ambarella document provides the following wireless connectivity API modules:

Module	Chapter	Description
Design and Configuration	Chapter 2	API design, generalized configuration and setting commands
System Commands	Chapter 3	Commands to manage camera system (e.g., format SD cards)
Session Controls	Chapter 4	Commands to initiate/terminate a session from a handheld device to a camera
Video Controls	Chapter 5	Commands to control video functionalities
Photo Controls	Chapter 6	Commands to control photographic functionalities
File System	Chapter 7	Commands to manage files (e.g., delete files)
WiFi	Chapter 8	Commands to restart the WiFi module
Notification	Chapter 10	Notifications from the Camera to the handheld
Media	Chapter 9	Commands associated with media (photos and video) files
Query	Chapter 11	Query commands sent from the camera to the handheld
Keywords	Chapter 12	List of JSON keywords
System Errors	Chapter 13	Key to error code returns

Table 1-1. Overview of the API Modules.

Readers of this document are assumed to be familiar with Ambarella wireless connectivity camera chips. Please contact an Ambarella representative for a list of related API documents and/or information on particular chips.

1.3 Overview: JSON Commands

The Remote Control API commands have uncommon flexibility due to their underlying structure and the extensible nature of the JavaScript Object Notation (JSON) language; i.e., any handheld capable of issuing JSON commands over a wireless connection (both WiFi and Bluetooth at this writing) can control one or more Ambarella cameras. For comprehensive detail on JSON syntax, please refer to http://json.org.

With the JSON syntax, parameter names are not hard-wired. The JSON parameters are generalized arguments, and the user can parse the content of a camera configuration file (i.e., the particular parameter names and menus on a given camera) and utilize them in the remote control commands.

JSON commands from the handheld to the Camera follow simple **key:value** pairs. Each pair is separated by a comma "," and each **key** is in the form of a string wrapped in quotation marks, "." The **value** may take the form of alphanumeric characters. For example,

{"token": 0, "msg_id": 1}

The arguments in the commands above are defined:

- **token** - Session token. Only one handheld can connect to one camera at a time. The token number will change with every session.
- **msg_id** - Command number. This is the number assigned to a particular remote command.

Please see Chapter 2 for additional background and general API forms used throughout this document.

1.4 Overview: Glossary

This document uses "handheld" to refer to portable devices such as smartphones (e.g., Apple iPhone, Samsung Galaxy) or tablets (e.g., iPad, Google Nexus) with wireless connectivity (e.g., WiFi and Bluetooth) capable of sending and receiving JSON messages and transferring data via a disparate connection.

2 Design Summary and Configuration API

2.1 Design: Overview

This chapter provides background information on the API design, including APIs used for the configuration of the parameter:

2.2 Design: Motivation

The Remote Control API structure is designed with two goals in mind:

- The primary goal is to establish seamless wireless communication between a handheld device and an Ambarella wireless-connectivity connected camera product.
- The second goal is to provide enough built-in flexibility to allow a single handheld application to control any Ambarella wireless connectivity product, regardless of differences in versions or capabilities.

2.3 Design: Setup

The principal requirement for establishing straightforward wireless communication between handhelds and Ambarella wireless-connectivity connected camera products is a simple and yet lightweight messaging protocol. The JSON (JavaScript Object Notation) protocol has been selected for this purpose.

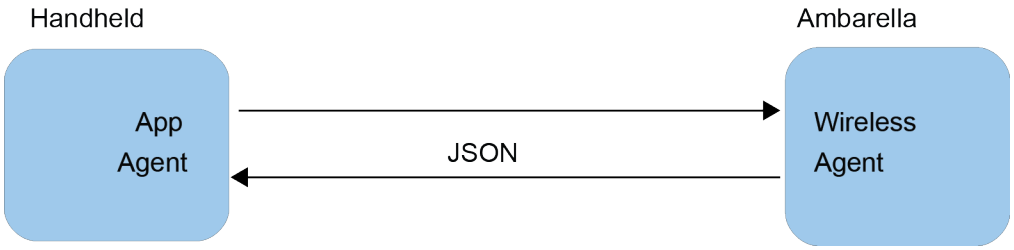

Figure 2-1. Communication between a Handheld Device and an Ambarella Connected Camera with JSON.

JSON allows a simple exchange of commands and return values with very few restrictions:

- Only single nested pairings of **key:values** are permitted. Nest pairings such as **{key1: {key2: value2}}**, for example, are not permitted. In addition, the use of alphanumeric characters is required.
- Numeric values can be decimal or eight-digit hexadecimal using the 0x prefix.

An Ambarella product can accommodate a connection to a single handheld. A handheld may have connections to multiple Ambarella products depending on the application implementation. Connections are session-based and each session has an associated token (**TokenNumber**). Sessions are created by the handheld sending an **AMBA_START_SESSION** request to the listening channel of the Ambarella wireless connectivity device. Once a session is successfully created, the assigned **TokenNumber** must be used with all subsequent commands. Where appropriate, commands result in a return value (**rval**) that signals the status of the command execution.

The handheld should be vigilant about receiving notifications (Chapter 10) from the camera. The camera sends notification to the handheld when manual camera operations take place. There may be times when the camera will need to disconnect the connection with the handheld in order to handle an internal task of higher priority. Once the session has been terminated, the handheld must create a new session to re-establish the connection within the camera.

HDMI-related operations, for example, are assigned a higher priority than a remote command. If the camera is connected to an HDMI device, the camera will refuse an **AMBA_START_SESSION** request and will terminate a running remote session (if one exists) by notifying the handheld of the HDMI insertion event.

Apart from notifications from the handheld, the Ambarella wireless-connectivity device will respond with JSON-based messages for every command sent from the handheld.

2.4 Design: Wireless Connectivity Distinctions

Wireless connectivity can be implemented in a variety of ways. WiFi and Bluetooth are supported by the Remote Control API currently. From a camera command viewpoint, the Remote Control API uses the same commands for WiFi and Bluetooth, making no distinction between them.

From a connection operational view, however, there are distinctions between wireless connectivity sources. Most importantly, some wireless connectivity (e.g., Bluetooth) protocols do not operate with TCP/IP. While Bluetooth has TCP/IP at its disposal via PAN (Personal Area Network) profiles, simultaneous Bluetooth PAN and LTE/4G limits most smartphone access to the greater internet at large, which makes PAN an unattractive option.

For these reasons, Bluetooth operates two Serial Port Profile (SPP) connections, one for commands and the other for data. By contrast, on TCP/IP-enabled infrastructure (e.g., WiFi) the commands are exchanged over the predetermined port 7878, and data are transmitted using port 8787.

The last wireless-connectivity distinction worth noting is bookkeeping. Bluetooth is a self-contained cable replacement protocol; whereas more configurations are necessary to support WiFi, especially when the camera is in Access Point mode. Section 2.8 of this document addresses Access Point maintenance configuration uniquely relevant to WiFi.

2.5 Design: Command Categorization

Using JSON over TCP/IP to port 7878 with WiFi and over SPP with Bluetooth provides a lightweight communication infrastructure between the handheld and the camera, as mentioned in the previous section. Therefore, achieving the second goal of allowing a single application on the handheld to control any wireless-connectivity Ambarella product becomes a matter of introducing flexibility (i.e., current and future).

Creating a unifying API that allows a single handheld application to remotely control all wireless-connectivity Ambarella products, spanning different camera capabilities and codes, is accomplished by separating all commands into two categories. The first category encompasses functions that do not vary from camera to camera. The second category consists of the variables that differ depending on the particular camera chosen.

There are certain functions present in every digital camera, whether the output is video or still images. These commands that do not vary from camera to camera include: start/stop record, shutter, file management, and so on. These functions are associated with fixed commands in the Remote Control API. The cameras themselves may execute these unchanging commands in an internally varying manner; however, the API and the command outcome is universal across all wireless-connectivity Ambarella products.

The variables that differ between cameras include features and capabilities such as resolutions and capture modes. Commands related to features and capabilities are categorized as settings and are handled with flexible generalized APIs (i.e., rather than assigning fixed commands that may have to be altered at a later date).

As camera technology progresses over time, features and capabilities improve from product to product and sometimes within the same product from version to version. As a result, the process of keeping the handheld application up to date with the latest features and capabilities can present a significant challenge. The Remote Control API has implemented Configuration files (Section 2.6) as a generalized solution to help address this challenge.

2.6 Design: Configuration of Settings and Capabilities

Using Remote Control API commands, the handheld treats the camera itself as a command / settings repository, as the handheld can access the camera and directly retrieve / set all of the parameter menus and options used to describe camera features and capabilities.

The camera maintains configuration settings and their associated values in a configuration database (Section 2.7), which is accessible by the handheld application through the use of **AMBA_SET_SETTING** (Section 3.2.2) and **AMBA_GET_SETTING** (Section 3.2.1) commands. For every parameter settable by the **AMBA_SET_SETTING** command, there is a corresponding **AMBA_GET_SETTING** command that can access its current value.

In summary, the camera configuration database provides a means to convey additional information from the camera to the handheld (e.g., present setting values, ranges of values for each setting, and the permission for each setting). As the handheld application retrieves these updates, it can configure and display the latest information to the user.

The configuration database is divided into tabs (or sections) for both video and photo. An important advantage of this type of configuration access is that, as camera technology progresses, additional tabs and entries can be added with the assurance that the handheld application can continue to use the same configuration management rules it uses today.

The next section introduces the commands to change and retrieve the parameters in a configuration.

2.7 Design: Configuration Data

Commands such as **AMBA_GET_ALL_CURRENT_SETTINGS** (Section 3.2.3), retrieve all the parameters' present values. **AMBA_GET_SETTING** (Section 3.2.1) retrieves a single specified parameter value. It is the responsibility of the handheld to use these two commands to synchronize its perception of the camera's state to the actual camera state.

2.8 Design: WiFi Configuration

Modifying WiFi settings using the WiFi channel itself increases the likelihood of instability. Therefore, WiFi settings are managed using an ASCII file via a PC or laptop, as part of the "out of box" (OOB) experience.

The `wifi.config` file must be located in the `MISC` directory:

```
/-
  | /MISC
              |/wifi.conf
  |/DCIM
              |/100MEDIA
```

A sample `wifi.conf` file is shown below.

```
##### Wifi configuration file
## Empty lines and lines starting with # are ignored
# ap: SoftAP mode
# sta: Station mode
# p2p: Enable Wifi Direct Support for peer-to-peer connectibity
WIFI_MODE=ap
# SSID (1 ~ 32 characters)
AP_SSID=NameOfCameraAP
# Passphrase (8 ~ 63 characters)

AP_PASSWD=Passphrase
# If you say yes here, the phassphrase setting will be ignored
AP_PUBLIC=no
# IP address
LOCAL_IP=192.168.42.1
# IP subnet mask
LOCAL_NETMASK=255.255.255.0
# IP pool starting address of DHCP server
DHCP_IP_START=192.168.42.2
# IP pool end address of DHCP server
DHCP_IP_END=192.168.42.4
# Wifi channel number, set 0 to use Auto Channel Selection
AP_CHANNEL=0
```

```
# Maximum number of stations allowed in station table
AP_MAXSTA=3
##### Advanced WPA configuration ##########################################
## Do not modify this section unless you are certain
# This is a bit field, bit0 = WPA, bit1 = WPA2
WPA_VERSION=2
# This is a space separated list of encryption algorithms, TKIP CCMP
WPA_PAIRWISE=CCMP
```

The following table provides an explanation of relevant WiFi configuration parameters:

Parameter	Definition
WIFI_MODE	Specifies the operating mode. This parameter should be set to one of the following values: `AP, STA or P2P`.
AP_SSID	This parameter should be set to the name of the camera.
AP_PASSWD	This pass-phrase should be kept secret to prevent unauthorized handheld access to the camera.
AP_PUBLIC	This parameter should be set to `no` unless public access to the camera is desired.
LOCAL_IP	This parameter should be set to `192.168.42.1`.
LOCAL_NETMASK	This parameter should be set to `255.255.255.0`.
DHCP_IP_END	This address is dependent on the parameter `AP_MAXSTA`. The default value of this address is `192.168.42.4` which is the gateway address `192.168.42.1` incremented by 3 (the default number of handheld devices permitted to connect to the camera).
AP_CHANNEL	When set to `0`, Automatic Channel Selection will select the least congested channel from 1, 6 and 11.
AP_MAXSTA	Total number of handheld devices that are permitted to connect to the camera. Three (`3`) is the default value. Note that only one handheld is able to control the camera at any given time.
WPA_VERSION	Specifies the WPA version
WPA_PAIRWISE	Specifies the algorithm for WPA

Table 2-1. WiFi Configuration Parameters and Definitions.

2.9 Design: Packet and Packet Headers

As noted, commands flowing on the command-channel follow JSON convention. Senders and receivers, by parsing according to JSON rules, can comprehend the content of the particular JSON item. The packet format and headers need to fulfill the same essential function for the items passing back and forth in the data channel. Please note, however, that this document will not specify the packet format and header information. Instead, in recognition of the customized nature of packet formats and headers for each customer, these will be defined in customer-specific documentation.

Some customers will choose to employ standardized packet formats such as TS (MPEG Transport Stream), while others may elect to circumvent royalty payments associated with common standards and improvise a customized packet header and format method.

When developing custom packet headers/formats, please take into account the following:

- Magic numbers to serve as start/stop code – Only data bookended by start and stop codes will be considered valid.
- Version of packet – Be sure to account for changes in packet versions.
- Payload content type.
- Payload size.
- Parent-child relationship – Payload may contain associated thumbnails to video and photo and vice versa.
- Data integrity checks, such as **md5sum**.

2.10 Design: Data Verification

md5sums are used for verification checks. The format of **md5sums** is alphanumeric and case-insensitive.

3 System Commands

3.1 System Commands: Overview

This section describes the API interface involved in camera system commands.

3.2 System Commands: List

This section lists the API System Commands 0x000000XX.

- AMBA_GET_SETTING
- AMBA_SET_SETTING
- AMBA_GET_ALL_CURRENT_SETTINGS
- AMBA_GET_SINGLE_SETTING_OPTIONS
- AMBA_FORMAT
- AMBA_GET_SPACE
- AMBA_GET_NUMB_FILES
- AMBA_NOTIFICATION
- AMBA_BURNIN_FW
- AMBA_NOTIFICATION > FW_UPGRADE_COMPLETE
- AMBA_PUT_GPS_INFO
- AMBA_GET_DEVICEINFO
- AMBA_SET_PASSWD
- AMBA_POWER_MANAGE
- AMBA_GET_BATTERY_LEVEL
- AMBA_ZOOM
- AMBA_ZOOM_INFO
- AMBA_CHANGE_BITRATE

3.2.1 AMBA_GET_SETTING

Message ID:

AMBA_GET_SETTING msg_id: 0x00000001(1)

API Overview:

- Type: Setting
- Description: This generalized API is used to retrieve the current value for a single camera parameter within the camera parameter database (Section 2.8). This API can be used for all camera parameters.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1	Remote command number (values must be decimal)
type	ParameterName	Name of the required parameter

*Table 3-1. Arguments for System Command API **AMBA_GET_SETTING**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	1	The **rval** is the response to this remote command
type	ParameterName	Name of the parameter
param	ParameterCurrentValue	Parameter specific details

*Table 3-2. Returns for System Command API **AMBA_GET_SETTING**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 1, "type": "video_resolution"}**
Successful return: **{"rval": 0, "msg_id": 1, "type": "video_resolution", "param":"1920x1080 30p 16:9"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"
- Chapter Appendix 5 "Naming Rule of Files in the Default Camera after API 4.1.0"

3.2.2 AMBA_SET_SETTING

Message ID:

AMBA_SET_SETTING msg_id: 0x00000002(2)

API Overview:

- Type: Setting (should be amenable to the selected Camera capabilities).
- Description: This is a generalized API to set individual camera parameter values within the parameter configuration database (Section 2.8). Set command examples in this document include:
  - Video quality and resolution (Chapter 5)
  - Photo capture mode and quality (Chapter 6)

  The **AMBA_SET_SETTING** command is also used to modify internal camera system components such as the system clock.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	ParameterName	Name of the parameter to set
param	ParameterValueToChange	Parameter specific details

*Table 3-3. Arguments for System Command API **AMBA_SET_SETTING**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	ParameterName	Confirmation of changed parameter name (as needed)
settable	settable option	Options (as needed) associated with or "coupled" to the selected **param.** Settable coupled parameters can be edited by the handheld.
readonly	read out	Read-only options (as needed) associated with or "coupled" to the selected **param.** These coupled parameters cannot be edited by the handheld.

*Table 3-4. Returns for System Command API **AMBA_SET_SETTING**.*

- It is the responsibility of the handheld application to properly decode and internally update the return values, especially if the changed parameter value results in other coupled parameters altering their values.

- The camera notifies the handheld of additional changes to the coupled parameter via the following syntax (Note that the separator character "**|**" is used to separate two coupled parameters in this code): **{"rval": 0, "msg_id": 2, "type": ParameterName, "settable":"param1_tab;param1_name;param1_value|param2_tab;param2_name;param2_value", "readonly":"param3_tab;param3_name;param3_value"}.** Please refer to http://www.json.org/ for comprehensive rules and syntax.
- The return separates parameters according to permission level: settable versus read-only. In the above example, the settable **key** has the **value** in its pairing of two settable parameters, **param1** and **param2**. **param1** is listed in its hierarchy of **tab;name;value** via the separator character semicolon, "**;**". **param1** and **param2** are divided by the separator character "**|**".
- The handheld application must properly parse these return values and update its internal bookkeeping in order for the camera and the handheld application to remain in sync.

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 2, "type": ParameterName, "param": ParameterValueToChange}**
- Standard return: **{"rval": 0, "msg_id": 2, "type": ParameterName, }**
- Return with coupled parameters: **{"rval": 0, "msg_id": 2, "type": ParameterName, "settable":"param1_tab;param1_name;param1_value|param2_tab;param2_name;param2_value", "readonly":"param3_tab;param3_name;param3_value"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.3 AMBA_GET_ALL_CURRENT_SETTINGS

Message ID:

AMBA_GET_ALL_CURRENT_SETTINGS msg_id: 0x00000003(3)

API Overview:

- Type: Setting
- Description:
  - This generalized API is used to retrieve current values for all camera parameters. This API can be used with all camera parameters.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	3	Remote command number (values must be decimal)

*Table 3-5. Arguments for System Command API **AMBA_GET_ALL_CURRENT_SETTINGS**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	3	The **rval** is the response to this remote command
param	[array of key:value pairs]	An array of all the **key:value** pairs representing all the current settings and their values.

*Table 3-6. Returns for System Command API **AMBA_GET_ALL_CURRENT_SETTINGS**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 3}**
Successful return: **{"rval": 0, "msg_id": 3, "param": [{"Video_Resolution":"1920x1080 30P 16:9"}, {"Video_Quality":Sfine} ....]}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.4 AMBA_GET_SINGLE_SETTING_OPTIONS

Message ID:

AMBA_GET_SINGLE_SETTING_OPTIONS msg_id: 0x00000009(9)

API Overview:

- Type: Setting as described in (Section 2.6) Design: Configuration of Settings and Capabilities.
- Description:
  - The **AMBA_GET_SETTING** API (Section 3.2.2) is used to retrieve all types of setting value(s). If the command **AMBA_GET_ALL_CURRENT SETTINGS** retrieves a snapshot of all current values for all parameters, this command **AMBA_GET_SINGLE_SETTING_OPTIONS** retrieves all of the possible options for ONE SINGLE SETTING. This should enable the handheld application to display all possible options with, for instance, a pull down menu.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION**
msg_id	9	Remote command number (values must be decimal)
param	<single setting string>	Name of setting

*Table 3-7. Arguments for System Command API **AMBA_GET_SINGLE_SETTING_OPTIONS**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	9	The **rval** is the response to this remote command
permission	<permission type>	Permission types: “readonly” and “settable”
param	<single setting string>	Name of setting
options	<array of options>	Options string will be all of the selectable values for this setting. Each selectable value is separated by a comma “,”.

*Table 3-8. Returns for System Command API **AMBA_GET_SINGLE_SETTING_OPTIONS**.*

Example:

Arguments to retrieve the camera’s clock and date: **{“token”: TokenNumber, “msg_id”: 9, “param”:“video_resolution”}**
Successful return: **{“rval”: 0, “msg_id”: 9, “permission” : ”settable”, “param”: “video_resolution”, “options”:[“1920x1080 60P 16:9” , ”1920x1080 30P 16:9” ,”1600x1200 60P 4:3” , ”1280x720 60P 16:9” ,”1280x720 30P 16:9” , ”848x480 60P 16:9” , ”848x480 30P 16:9” , ”1280x720 120P 16:9” , ”848x480 120P 16:9”]}**

See Also:

- Section 2.6 “Design: Configuration of Settings and Capabilities”
- Section 3.2.2 “AMBA_SET_SETTING”
- Chapter 13 “System Errors”

3.2.5 AMBA_FORMAT

Message ID:

AMBA_FORMAT msg_id: 0x00000004(4)

API Overview:

- Type: Function.
- Description: This API is used to format the camera SD card.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	4	Remote command number (values must be decimal)
param	SD card	Alphabet corresponding to SD card

*Table 3-9. Arguments for System Command API **AMBA_FORMAT**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	4	The **rval** is the response to this remote command

*Table 3-10. Returns for System Command API **AMBA_FORMAT**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 4, "param":"D:"}**
Successful return: **{"rval": 0 , "msg_id": 4}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.6 AMBA_GET_SPACE

Message ID:

AMBA_GET_SPACE msg_id: 0x00000005(5)

API Overview:

- Type: Function.
- Description: This API is used to retrieve the camera's total memory space and total free space that is remaining.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	5	Remote command number (values must be decimal)
type	<SpaceTypeInString>	Space type: "total": Total space "free": Free space remaining

*Table 3-11. Arguments for System Command API **AMBA_GET_SPACE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	5	The **rval** is the response to this remote command
param	<NumbOfKBytes>	Size of required information (KBytes)

*Table 3-12. Returns for System Command API **AMBA_GET_SPACE**.*

Example:

Arguments: **:{"token":TokenNumber,"msg_id":5,"type":"total"}** Request amount of free space
Successful return: **{"rval":0,"msg_id":5,"param":4}** 4K free space

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.7 AMBA_GET_NUMB_FILES

Message ID:

AMBA_GET_NUMB_FILES msg_id: 0x00000006(6)

API Overview:

- Type: Function.
- Description: This API is used to retrieve the number of files. Types include: total, photo, and video.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	6	Remote command number (values must be decimal)
type	<FileTypeString>	Type of file: "total": Total files "photo": Photo files "video": Video files

*Table 3-13. Arguments for System Command API **AMBA_GET_NUMB_FILES**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	6	The **rval** is the response to this remote command
param	<NumbOfFiles>	Number of files

*Table 3-14. Returns for System Command API **AMBA_GET_NUMB_FILES**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 6, "type": "video"}** Request total number of files
Successful return: **{"rval": 0, "msg_id": 6, "param": 527}** 527 files

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.8 AMBA_NOTIFICATION

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: Notifications are sent from the camera to the connected handheld to notify it of events, such as manual button presses and task completions (e.g., **AMBA_GET_FILE**). Chapter 10 discusses Notifications. Notification types are uniquely defined by Ambarella API keywords (Chapter 12) .
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	<Notification Keyword>	Notification type

*Table 3-15. Arguments for System Command API **AMBA_NOTIFICATION**.*

Returns:

None. A notification is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the changes in camera state.

Example:

- Arguments: **{"msg_id": 7, "type":<Notification Keyword> }**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

3.2.9 AMBA_BURNIN_FW

Message ID:

AMBA_BURNIN_FW msg_id: 0x00000008(8)

API Overview:

- Type: Function
- Description: This API programs a camera after an upgraded firmware (FW) image has been uploaded. This step represents the final step of the FW upgrade process.

  Caution should be taken not to wait too long between the completion of the FW upgrade file upload and **AMBA_BURNIN_FW** as the battery level may decrease below operational levels. Before the **AMBA_BURNIN_FW** starts, the camera will check the appropriate battery levels.

  In some designs of the camera, at the completion of the command, the camera will be automatically rebooted, and if the FW upgrade is successful, the notification **AMBA_NOTIFICATION > FW_UPGRADE_COMPLETE** will be sent to the handheld before rebooting. However, it is based on the design of camera whether the notification will be sent to the handheld or not.

- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	8	Remote command number (values must be decimal)
param	File Name	Full path of the firmware file

*Table 3-16. Arguments for System Command API **AMBA_BURNIN_FW**.*

Returns:

Key	Value	Description
rval	< 0	0: Success <0: Error as defined in Chapter 13
msg_id	8	The rval is the response to this remote command

*Table 3-17. Arguments for System Command API **AMBA_BURNIN_FW**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 8, "param":"/tmp/fuse_d/firmware.bin" }**
- Successful return: **{"rval": 0, "msg_id": 8} //success**
- Error return: **{"rval": -13, "msg_id": 8} //file does not exist**

See Also:

- Section 4.2.1 “AMBA_START_SESSION”
- Chapter 13 “System Errors”

3.2.10 AMBA_NOTIFICATION > FW_UPGRADE_COMPLETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x0000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld the command **AMBA_BURNIN_FW** has completed. Immediately after sending this notification, the camera will reboot for the newly burned FW to take effect. Please allow sufficient time for the camera to reboot before the handheld attempts to reconnect to the camera.
- Direction: From the Camera to the handheld

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"fw_upgrade_complete"	Notification of successful transmission

*Table 3-18. Arguments for System Command Notification **AMBA_NOTIFICATION > FW_UPGRADE_COMPLETE**.*

Returns

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to re-establish connection to the camera after the reboot especially if the handheld is in WiFi STA mode.

Example:

- Arguments: **{"msg_id": 7, "type":"fw_upgrade_complete"}**
- Chapter 4 "Session Controls"

3.2.11 AMBA_PUT_GPS_INFO

Message ID:

AMBA_PUT_GPS_INFO msg_id: 0x0000000a(10)

API Overview:

- Type: Function.
- Description: This API sends GPS-related information to the camera. When the camera is recording and **EXT_GPS** is enabled, this information will be saved to the video clip. The handheld device should send GPS information at a frequency of 1/s in order to increase the accuracy of the GPS data.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	10	Remote command number (values must be decimal)
param	<GPS info>	GPS information to be saved into clip

*Table 3-19. Arguments for System Command API **AMBA_PUT_GPS_INFO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	10	The **rval** is the response to this remote command

*Table 3-20. Returns for System Command API **AMBA_PUTGPS_INFO**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 10,"param" }"X0000.0000Y0000.0000Z0000.0000G0000.0000$GPRMC,000125,V,,,,,000.0,,280908,002.1,N*71~" } //GPS info**
- Successful return: **{"rval": 0, "msg_id": 10} //success**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.12 AMBA_GET_DEVICEINFO

Message ID:

AMBA_GET_DEVICEINFO msg_id: 0x0000000b(11)

API Overview:

- Type: Function
- Description: This API is used to retrieve information related to the camera device itself, such as the model name, logo path, firmware version, and camera application type.
- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	11	Remote command number (values must be decimal)

*Table 3-21. Arguments for System Command API **AMBA_GET_DEVICEINFO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	11	The message ID of this command.
brand	<Brand name>	Camera vendor's brand name
model	<Mode name>	Camera model name
event_folder	<event folder path>	Full path of event directory. Note that the phone-App should use default value (i.e. /tmp/SD0/EVENT/) if **AMBA_GET_DEVICEINFO** -> **event_folder** is absent in **AMBA_GET_DEVICEINFO**, and the phone-App should be aware that this directory may not exist if the camera has never executed the event record before.
sec_posfix	<posfix of secondary file>	Posfix of the secondary file
naming_rule	[array of key:value pairs]	This key is used to provide naming rules for files from the camera to the handheld device. The handheld device can design algorithms based on these limitations. The camera can design it based on any naming rule as there are no other known limitations. The user is requested to check if the application in the handheld can analyze the limitations.
http	<status of http service>	Disable: The http service is not supported. Enable: The http service is supported.
auth	<status of authentication>	On: Password needed for the handheld to acquire the session and change the password Off: Password not needed for the handheld to acquire the session

Key	Value	Description
chip	<chip version>	Camera version: A5s A7L
app_type	<camera type>	Camera application type: Car Sports Wearable
fw_ver	<firmware version>	Camera firmware version
api_ver	<Remote API version>	The Remote API version supported by the current camera firmware. This version number is the same as the API document version that is implemented in the current camera firmware. For example, *AMBARELLA A-Series Wireless Connectivity API Remote Control, v2.8.1*.
logo	<logo file path>	Full path of logo file
media_folder	<media folder path>	Full path of media directory. Note that the phone-App should use the default value (i.e. /tmp/SD0/DCIM/) if **AMBA_GET_DEVICEINFO** -> **media_folder** is absent in the **AMBA_GET_DEVICEINFO**.
event_folder	<event folder path>	Full path of event directory. Note that the phone-App should use default value (i.e. /tmp/SD0/EVENT/) if **AMBA_GET_DEVICEINFO** -> **event_folder** is absent in **AMBA_GET_DEVICEINFO**, and the phone-App should be aware that this directory may not exist if the camera has never executed the event record before.
sec_posfix	<posfix of secondary file>	Posfix of the secondary file
naming_rule	[array of key:value pairs]	This key is used to provide naming rules for files from the camera to the handheld device. The handheld device can design algorithms based on these limitations. The camera can design it based on any naming rule as there are no other known limitations. The user is requested to check if the application in the handheld can analyze the limitations.
http	<status of http service>	Disable: The http service is not supported. Enable: The http service is supported.
auth	<status of authentication>	On: Password needed for the handheld to acquire the session and change the password Off: Password not needed for the handheld to acquire the session

*Table 3-22. Returns for System Command API **AMBA_GET_DEVICEINFO**.*

Example:

In A7L: (Before API 4.1.0)

- Arguments: **{"token": TokenNumber, "msg_id": 11}**
- Successful return: **{"rval": 0, "msg_id": 11, "brand":"ambarella", "model":"vantage", "chip":"a7l", "app_type":"car", "fw_ver":"Mon Dec 9 20:28:27 CST 2013", "api_ver":"2.7.10", "logo":"/tmp/fuse_z/logo.bin", "media_folder":"/tmp/fuse_d/DCIM/", "event_folder":"/tmp/fuse_d/EVENT", "sec_postfix":"_thm", "http":"disable", "auth":"off"}**

In A12: (After API 4.1.0)

- Arguments: {**"token": TokenNumber, "msg_id": 11**}
- Successful return: **{"rval": 0, "msg_id": 11, "brand":"ambarella", "model":"vantage", "chip":"a12", "app_type":"car", "fw_ver":"Mon Dec 9 20:28:27 CST 2013", "api_ver":"4.1.0", "logo":"/tmp/fuse_z/logo.bin", "media_folder":"/tmp/SD0/DCIM/", "event_folder":"/tmp/SD0/EVENT", "naming_rule": [{"type": "video", "main_section": [0,6], "sensor_section":[6,1], "stream_section":[7,1] }, {"type": "photo", "main_section": [0,6],"offset_section":[6,2]}], "http":"disable", "auth":"off"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"
- Appendix 5 "Naming Rule of Files in the Default Camera after API 4.1.0"

3.2.13 AMBA_SET_PASSWD

Message ID:

AMBA_SET_PASSWD msg_id: 0x00000107(263)

API Overview:

- Type: Function.
- Description: This API enables/disables and changes the password which is carried with **AMBA_START_SESSION** for authentication.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	263	Remote command number (values must be decimal)
auth	on/off	Enable/Disable authentication
old_passwd	<password string>	Old password used for authentication Note: The encryption method of password is left to camera firmware and handheld program developer. One thing to note is that password should be in the string format. In case the encrypted password contains non-string characters, it is recommended to encode the encrypted password into a base64 string.
passwd	<password string>	New password

*Table 3-23. Arguments for System Command API **AMBA_SET_PASSWD**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	263	The **rval** is the response to this remote command

*Table 3-24. Returns for System Command API **AMBA_SET_PASSWD**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id":263, "auth": "on", "old_passwd":"old_passwd_base64", "passwd":"passwd_base64"}**
- Successful return: **{"rval": 0, "msg_id": 263}**

See Also:

- Section 4.2.1 “AMBA_START_SESSION”
- Chapter 13 “System Errors”

3.2.14 AMBA_POWER_MANAGE

Message ID:

AMBA_POWER_MANAGE msg_id: 0x0000000c(12)

API Overview:

- Type: Function.
- Description: This API is used to either switch the camera operating mode to standby mode or to power-off the camera device.
- At the completion of this command, the camera will switch to the specified mode or will power-off. If the operation is successful, the Notification **AMBA_NOTIFICATION>POWER_MODE_CHANGE** will be sent to the handheld.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	12	Remote command number (values must be decimal)
param	<mode>	The requested mode change: "cam_stb": Standby both RTOS and Linux. "cam_off": Power-off both RTOS and Linux.

*Table 3-25. Arguments for System Command API **AMBA_POWER_MANAGE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	12	The **rval** is the response to this remote command

*Table 3-26. Returns for System Command API **AMBA_POWER_MANAGE**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 12,"param":"cam_stb" }**
- Successful return: **{"rval": 0, "msg_id": 12}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.15 AMBA_GET_BATTERY_LEVEL

Message ID:

AMBA_GET_BATTERY_LEVEL msg_id: 0x0000000d(13)

API Overview:

- Type: Function.
- Description: This API is used to retrieve the battery status. The camera will return the current battery level in terms of percentage.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	13	Remote command number (values must be decimal)

*Table 3-27. Arguments for System Command API **AMBA_GET_BATTERY_LEVEL**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	13	The **rval** is the response to this remote command.
type	<type of power source>	Indicate the current power source of the camera. "battery": Powered by battery. "adapter": Adapter is connected.
param	<battery_level>	Camera battery level percentage remaining

*Table 3-28. Returns for System Command API **AMBA_GET_BATTERY_LEVEL**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 13 }**
- Successful return: **{"rval": 0, "msg_id": 13, "type" : "adapter", "param": "75" }**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

3.2.16 AMBA_ZOOM

Message ID:

AMBA_ZOOM msg_id: 0x0000000e(14)

API Overview:

- Type: Function.
- Description: This command is used to zoom in, zoom out, reset zoom level, or get current zoom level.
- Direction: From the handheld to the Camera.

Arguments:

Key	Example Values	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	14	Remote command number (values must be decimal)
type	<type>	jump: To zoom to specific steps directly. fast: to zoom to specific steps smoothly in fast speed normal: To zoom to specific steps smoothly in normal speed. slow: To zoom to specific steps smoothly in slow speed.
param	<level>	Integer number, to indicate digital zoom level. Handheld device can retrieve the max room info from **AMBA_ZOOM_INFO**.

*Table 3-29. Arguments for System Command API **AMBA_ZOOM**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	14	The **rval** is the response to this remote command.

*Table 3-30. Returns for System Command API **AMBA_ZOOM**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 14, "type" : "jump" , "param" : "500"}**
- Successful return: **{"rval": 0, "msg_id": 14} //success**
- Error return:
- **{"rval": -21, "msg_id": 772 }** //system is busy(recording)
- **{"rval": -23, "msg_id": 772 }** //device don't support this operation
- **{"rval": -25, "msg_id": 772 }** //invalid param => illegal or param

See Also:

- Section 4.2.1 “AMBA_START_SESSION”
- Chapter 13 “System Errors”

3.2.17 AMBA_ZOOM_INFO

Message ID:

AMBAL_ZOOM_INFO msg_id: 0x0000000F(15)

API Overview:

- Type: Function.
- Description: This command is used to retrieve zoom information of the camera.
- Direction: From the handheld to the Camera.

Arguments:

Key	Example Values	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	15	Remote command number (values must be decimal)
type	<type>	"max:" The maximum zoom level allowed in this device. "current": The current zoom level in this device. "status": Indicate that zoom is "processing" or "idle".

*Table 3-31. Arguments for System Command API **AMBA_ZOOM_INFO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	15	The **rval** is the response to this remote command.
param	<"level">	The requested information While requesting the "max", this parameter is to represent the maximum steps of zoom operation. However, the camera will return "param" as 0 if the current resolution does not support the zoom operation.

*Table 3-32. Returns for System Command API **AMBA_ZOOM_INFO**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 15, "type":"max" }**
- Successful return: **{"rval": 0, "msg_id": 15, "param": "486"} //success , device supports zoom function on current resolution**
- Arguments: **{"token": TokenNumber, "msg_id": 15, "type":"max" }**
- Successful return: **{"rval": 0, "msg_id": 15, "param": "0"} //success, device doesn't support zoom function on current resolution**
- Arguments: **{"token": TokenNumber, "msg_id": 15, "type":"max" }**
- Successful return: **{"rval": -23, "msg_id": 15} //device doesn't support zoom function on all resolution**

- Error return:
- **{"rval": -21, "msg_id": 772 }** //system is busy(recording)
- **{"rval": -23, "msg_id": 772 }** //device doesn't support this operation
- **{"rval": -24, "msg_id": 772 }** //invalid type

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 3 "System Commands"

3.2.18 AMBA_CHANGE_BITRATE

AMBA_CHANGE_BITRATE msg_id: 0x00000010(16)

API Overview:

- Type: Function.
- Description: This API is used to change to birate of streaming.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	16	Remote command number (values must be decimal)
param	<bitrate in Kbits/second>	The bitrate of streaming video

*Table 3-33. Arguments for Session Control API **AMBA_CHANGE_BITRATE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	16	The **rval** is the response to this remote command.

*Table 3-34. Returns for Session Control API **AMBA_CHANGE_BITRATE**.*

Example:

Arguments: {**"token" : TokenNumber, "msg_id" :16**,**"param": "100"**}
// change bitrate to 100Kbits/sec
Successful return: {**"rval : 0, "msg_id" : 16**}
Error return:
{**"rval" : -23, "msg_id" : 16** } //device doesn't support this operation
{**rval : -25, "msg_id" : 16** } //invalid param => illegal birate

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

4 Session Controls

4.1 Session Controls: Overview

This section list the API interface involved in session management between the handheld and the camera.

4.2 Session Controls: List

This section lists the API Session Control Commands 0x000001XX.

- AMBA_START_SESSION
- AMBA_SET_CLNT_INFO
- AMBA_STOP_SESSION
- AMBA_RESETVF
- AMBA_STOP_VF

4.2.1 AMBA_START_SESSION

Message ID:

AMBA_START_SESSION msg_id: 0x00000101(257)

API Overview:

- Type: Function
- Description: This API is used to request the start of a session from the handheld to the camera.
- Direction: Handheld to the Camera.
- Returns: Return value (**rval**) and the session **TokenNumber** used in all subsequent command exchanges.

Arguments:

Key	Value	Description
token	0	Starting value of the session
msg_id	257	Remote command number (values must be decimal)
Passwd (Option)	<password string>	Password Note: The encryption method of the password is left to the camera firmware and handheld program developer. One thing to note is that password should be in string format. In case the encrypted password contains non-string character, it is recommended to encode the encrypted password into base64 string.

*Table 4-1. Arguments for Session Control API **AMBA_START_SESSION**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Successful session creation between the requesting handheld and the targeted Ambarella camera <0: Error as defined in Chapter 13
msg_id	257	The **rval** is the response to this remote command
param	**TokenNumber**	The returned **param** is **TokenNumber** (in integer value) of this session. Use this **TokenNumber** in all subsequent command exchanges between the handheld and the camera.

*Table 4-2. Returns for Session Control API **AMBA_START_SESSION**.*

Example 1:

Arguments: **{"token": 0, "msg_id": 257}**
Return: **{"rval": 0, "msg_id": 257, "param": TokenNumber}**

Example 2:

Arguments: **{“token”:O,“msg_id”: 257, “passwd”: “password”}**
Successful return: **{ “rval”: 0, “msg_id”:257, “param”: Tokennumber}**
Error return:

- **{“rval”: -3, “msg_id”: 257} //someone holds the session**
- **{“rval”: -29, “msg_id”: 257} //the password is wrong**

See Also:

- Chapter 13 “System Errors”

4.2.2 AMBA_SET_CLNT_INFO

Message ID:

AMBA_SET_CLNT_INFO msg_id: 0x00000105(261)

API Overview:

- Type: Function
- Description: This API sets the connection information of the active client. Handheld device is obligated to set connection information before creating the data socket connection. Camera will verify the address information with incoming data socket connection and reject it if the verification does not pass.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	261	Remote command number (values must be decimal)
type	<Transport protocol>	Transport protocol type: TCP: Use TCP/IP protocol to create data connection RFCOMM: Use rfcomm protocol to create data connection
param	<identity>	This identity is depend on transport protocol type and should be unique for each client. The value will be: IP address, such as “192.168.42.5”, if transport protocol type is “TCP”. MAC address, such as “a0:b1:c2:d3:e4:f5”, if transport protocol type is “RFCOMM”.

*Table 4-3. Arguments for Session Control API **AMBA_SET_CLNT_INFO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Successful session termination <0: Error as defined in Chapter 13
msg_id	261	The **rval** is the response to this remote command

*Table 4-4. Returns for Session Control API **AMBA_SET_CLNT_INFO**.*

Example:

For TCP:

Arguments: **{"token": TokenNumber, "msg_id":261, "type":"TCP","param": "192.168.42.5"}**
Successful return: **{"rval": 0 , "msg_id": 261}**

For RFCOMM:

Arguments: **{"token": TokenNumber, "msg_id":261, "type":"RFCOMM" ,"param": "a0:b1:c2:d3:e4:f5"}**
Successful return: **{ "rval": 0, "msg_id":261}**
Error return:

- {"rval": -25, "msg_id": 261} // the client identity is invalid
- {"rval": -23, "msg_id": 261} // device does not support this operation

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

4.2.3 AMBA_STOP_SESSION

Message ID:

AMBA_STOP_SESSION msg_id: 0x00000102(258)

API Overview:

- Type: Function
- Description: This API is used to request session termination between the handheld and the camera.
- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	258	Remote command number (values must be decimal)

*Table 4-5. Arguments for Session Control API **AMBA_STOP_SESSION**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Successful session termination <0: Error as defined in Chapter 11
msg_id	258	The **rval** is the response to this remote command

*Table 4-6. Returns for Session Control API **AMBA_STOP_SESSION**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 258}**
Successful return: Success: **{"rval": 0 , "msg_id": 258}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

4.2.4 AMBA_RESETVF

Message ID:

AMBA_RESETVF msg_id: 0x00000103(259)

API Overview:

- Type: Function
- Description:
  - This API is used to actively turn on (or reset, if already enabled) the Remote View Finder. This command will initialize the remote view finder if it is currently turned off.
  - This is an optional command, useful when the handheld requires an active restart of the view finder server.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	259	Remote command number (values must be decimal)
param	none_force (Optional)	If this parameter is set, the camera will use the current settings to turn-on the view finder. If the view finder is disabled, it will not be turned on. If this parameter is not set, the camera will force a reset of view finder settings to their default values and turn-on the view finder. This is an optional command.

*Table 4-7. Arguments for Session Control API **AMBA_RESETVF**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Successful restart of the viewfinder server <0: Error as defined in Chapter 13
msg_id	259	The **rval** is the response to this remote command

*Table 4-8. Returns for Session Control API **AMBA_RESETVF**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 259}**
Successful return: **{"rval": 0 , "msg_id": 259}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

Note:
Handheld device should send **AMBA_RESETVF** to enable the streaming if app_status is idle/photo_mode.
If capture_mode = precise_cont, which means that the camera is performing continuous capture operation. In this status, the handheld device has to send **AMBA_CONTINUE_CAPTURE_STOP** to stop the continuous capture operation if it wants to operate the camera and receive the streaming data.

4.2.5 AMBA_STOP_VF

Message ID:

AMBA_STOP_VF msg_id: 0x0000104(260)

API Overview:

- Type: Function
- Description: This API is used to stop the Remote View Finder.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	260	Remote command number (values must be decimal)

*Table 4-9. Arguments for Session Control API **AMBA_STOP_VF**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Successful stopping of the camera encoder <0: Error as defined in Chapter 13
msg_id	260	The **rval** is the response to this remote command.

*Table 4-10. Returns for Session Control API **AMBA_STOP_VF**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 260}**
Successful return: **{"rval": 0 , "msg_id": 260}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

5 Video Commands

5.1 Video Commands: Overview

This section discusses the API interface for controlling video functionality.

5.2 Video Commands: List

This section lists the API Video Commands 0x000002XX.

- AMBA_RECORD_START
- AMBA_RECORD_STOP
- AMBA_GET_RECORD_TIME
- AMBA_FORCE_SPLIT

5.2.1 AMBA_RECORD_START

Message ID:

AMBA_RECORD_START msg_id: 0x00000201(513)

API Overview:

- Type: Function.
- Description: This API is used to initiate camera video recording.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	513	Remote command number (values must be decimal)

*Table 5-1. Arguments for Video API **AMBA_RECORD_START**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	513	The **rval** is the response to this remote command.

*Table 5-2. Returns for Video API **AMBA_RECORD_START**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 513}**
Successful return: **{"rval": 0, "msg_id": 513}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

5.2.2 AMBA_RECORD_STOP

Message ID:

AMBA_RECORD_STOP msg_id: 0x00000202(514)

API Overview:

- Type: Function.
- Description: This API is used to terminate video recording on the camera..
- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	514	Remote command number (values must be decimal)

*Table 5-3. Arguments for Video API **AMBA_RECORD_STOP**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	514	The **rval** is the response to this remote command.
param	<saved file name>	File name including fully qualified path.

*Table 5-4. Returns for Video API **AMBA_RECORD_STOP**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 514}**
Successful return: **{"rval": 0, "msg_id": 514, "param":"/tmp/fuse_d/DCIM/100MEDIA/AMBA0001.mp4"}**

See Also:

- (Section 4.2.1) AMBA_START_SESSION
- Chapter 13 "System Errors"

5.2.3 AMBA_GET_RECORD_TIME

Message ID:

AMBA_GET_RECORD_TIME msg_id: 0x00000203(515)

API Overview:

- Type: Function.
- Description: This API is used to retrieve the duration (in seconds) of the current video recording. This command is valid only while the camera is recording video.
- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	515	Remote command number (values must be decimal)

*Table 5-5. Arguments for Video API **AMBA_GET_RECORD_TIME**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	515	The **rval** is the response to this remote command.
param	<DurationInSec>	Current recording duration in seconds (integer type)

*Table 5-6. Returns for Video API **AMBA_GET_RECORD_TIME**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 515}**
Successful return: **{"rval": 0, "msg_id": 515, "param": 120}** Recording duration is 120 secs = 2 mins
Error return: **{"rval": -14, "msg_id": 515}** If the camera is not in record mode, **-14:INVALID_OPERATION** is returned.

See Also:

- (Section 4.2.1) AMBA_START_SESSION
- Chapter 13 "System Errors"

5.2.4 AMBA_FORCE_SPLIT

Message ID:

AMBA_FORCE_SPLIT msg_id: 0x00000204(516)

API Overview:

- Type: Function.
- Description: This API requests the camera to save the current recording clip immediately and continue recording using a new file. The return for this API will contain the full path filename of the clip, which can subsequently be used by the handheld device to retrieve the saved file via an **AMBA_GET_FILE** command.
- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	516	Remote command number (values must be decimal)

*Table 5-7. Arguments for Video API **AMBA_FORCE_SPLIT**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	516	The **rval** is the response to this remote command.
param	<saved file name>	File name including fully qualified path.

*Table 5-8. Returns for Video API **AMBA_FORCE_SPLIT**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 516}**
Successful return: **{"rval": 0, "msg_id": 516, "param": "/tmp/fuse_d/DCIM/100MEDIA/AMBA0005.mp4"}**
Error return**: {"rval": -14, "msg_id": 516}** If the camera is not in record mode**, -14:INVALID_OPERATION** is returned.

See Also:

- (Section 4.2.1) AMBA_START_SESSION
- Chapter 13 "System Errors"

See Also:

- Section 2.6 “Design: Configuration of Settings and Capabilities”
- Section 4.2.1 “AMBA_START_SESSION”
- Section 3.2.2 “AMBA_SET_SETTING”
- Chapter 13 “System Errors”

6 Photo Commands

6.1 Photo Commands: Overview

This section discusses the API interface for the controlling digital photographic functionality.

6.2 Photo Commands: List

This section lists the API Photo Commands 0x000003XX.

- AMBA_TAKE_PHOTO
- AMBA_NOTIFICATION > CONTINUE_BURST_COMPLETE
- AMBA_CONTINUE_CAPTURE_STOP

6.2.1 AMBA_TAKE_PHOTO

Message ID:

AMBA_TAKE_PHOTO msg_id: 0x00000301(769)

API Overview:

- Type: Function.
- Description: This API is used to capture a still image, begin a precise continue capture or start to perform burst continue capture depending on the value of **AMBA_GET_SETTING** -> **capture_mode**. If there is no **capture_mode** available in the **AMBA_GET_ALL_CURRENT_SETTING**, the camera only supports precise capture(single photo). Note that when capture_mode= "precise_cont" or "burst_cont" (continue capture), the hand-held device is obligated to send **AMBA_CONTINUE_CAPTURE_STOP** to stop the continue capture operation.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	769	Remote command number (values must be decimal)

*Table 6-1. Arguments for Photo API **AMBA_TAKE_PHOTO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	769	The **rval** is the response to this remote command.
param (option)	<saved file name>	File name including fully qualified path. Note: The time it takes to generate taken photo may differ largely due to different hardware configuration. To avoid long command respond time, some of the implementation might not return filename directly in response. They can choose not to return filename but send notification of "**photo_taken**" until taken photo is actually generated. Thus, hand-held device should expect "**photo_taken**" notification when this field is absent. Furthermore, this field will be absent if capture_mode = "precise_cont" or "burst_cont".

*Table 6-2. Returns for Photo API **AMBA_TAKE_PHOTO**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 769}**
Successful return:
capture_mode= "precise quality":
{"rval": 0, "msg_id": 769, "param": "/tmp/fuse_d/DCIM/100MEDIA/AMBA0005.jpg}
capture_mode="precise quality cont.", "burst quality cont."
{"rval":0, "msg_id":769} //means that continue capture has successfully begun
// or the take_photo operation success and notifications will be issued few seconds later
Error return:
{"rval":-14, "msg_id": 769} //already in capture operation
{"rval":-20, "msg_id": 769} //camera is recording and PIV(photo in video) is not allowed

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities""
- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

6.2.2 AMBA_NOTIFICATION > CONTINUE_BURST_COMPLETE

Message ID:

CONTINUE_BURST_COMPLETE msg_id: 0x0000007(7)

API Overview:

- Type: Notification.
- Description: This API is used to by the camera to signal to the handheld that the burst capture operation has completed.
- Direction: From the handheld to the Camera.

Arguments:

Key	Example Values	Description
msg_id	7	Remote command number (values must be decimal)
type	<continue_burst_complete>	To indicate that the operation of burst capture has completed.
param	<file path>	The file path of the last photo taken

*Table 6-3. Arguments for Photo API **AMBA_NOTIFICATION > CONTINUE_BURST_COMPLETE**.*

Returns:

None. This command is to be received and acted upon by the handheld.

Example:

Arguments: **{"msg_id":7,"type":"continue_burst_complete","param":"<filename of last photo>"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

6.2.3 AMBA_CONTINUE_CAPTURE_STOP

Message ID:

AMBA_CONTINUE_CAPTURE_STOP msg_id: 0x00000302(770)

API Overview:

- Type: This API is used to stop a precise continue capture operation.
- Description: This API is used to stop a precise continue capture operation.
- Direction: From the handheld to the Camera.

Arguments:

Key	Example Values	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	770	Remote command number (values must be decimal)

*Table 6-4. Arguments for Photo API **AMBA_CONTINUE_CAPTURE_STOP**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	770	The **rval** is the response to this remote command.

*Table 6-5. Returns for Photo API **AMBA_CONTINUE_CAPTURE_STOP**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 770}**
- Successful return: **{"rval": 0, "msg_id": 770}**
- Error return:

{"rval": -14, "msg_id": 770} //cannot do this, maybe camera is not doing Continue_Capture
{"rval": -21, "msg_id": 770} //system is busy (recording)
{"rval": -23, "msg_id": 770} //device doesn't support this operation
{"rval": -23, "msg_id": 770} //device doesn't support this operation

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

7 File System Commands

7.1 File System Commands: Overview

This section discusses the API interface involved in the access and maintenance of files and directories in the camera storage.

7.2 File System Commands: List

This section lists the API File System Commands 0x000005XX.

- AMBA_DEL_FILE
- AMBA_LS
- AMBA_CD
- AMBA_PWD
- AMBA_GET_FILE
- AMBA_CANCEL_FILE_XFER
- AMBA_NOTIFICATION > GET_FILE_COMPLETE
- AMBA_PUT_FILE
- AMBA_NOTIFICATION > PUT_FILE_COMPLETE
- AMBA_MKDIR

7.2.1 AMBA_DEL_FILE

Message ID:

AMBA_DEL_FILE msg_id: 0x00000501(1281)

API Overview:

- Type: Function.
- Description: This API is used to delete a file.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1281	Remote command number (values must be decimal)
param	file	File name including fully qualified path or in local directory

*Table 7-1. Arguments for File System Command API **AMBA_DEL_FILE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 12
msg_id	1281	The **rval** is the response to this remote command

*Table 7-2. Returns for File System Command API **AMBA_DEL_FILE**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 1281, "param": "DCIM/100MEDIA/AMBA0001.MP4"}**
Successful return: **{ "rval": 0, "msg_id": 1281 }**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.2 AMBA_LS

Message ID:

AMBA_LS msg_id: 0x00000502(1282)

API Overview:

- Type: Function.
- Description: This API lists the contents of the directory which is current or specified by the parameter, analogous to the command "`ls`". The command will return an array of content for the directory. An empty array ([ ]) is returned if the directory is empty. The returned array has tuple(s) as elements. In each element, the key is a file or a directory, which have a trailing "/". Timestamps follow the convention of YYYY-MM-DD HH:MM:SS. When **param** is specified, different information will be shown depending on the **param** value.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1282	Remote command number (values must be decimal)
Param (optional)	<file list option>	The file(or directory) names and created date of all files in the current directory will be returned if any parameter stated below is not specified. The user can specify the following parameters: 1. The full path of specified directory. (ex. /tmp/fuse_d/DCIM/100MEDIA/) 2."-D": The created date of the file 3. "-S": The size of the file

*Table 7-3. Arguments for File System Command API **AMBA_LS**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success -25 (INVALID_PARAM): Invalid option -26 (INVALID_PATH): No such directory.
msg_id	1282	The **rval** is the response to this remote command.
listing	Contents of the cur rent or specified directory	Contents of the current directory

*Table 7-4. Returns for File System Command API **AMBA_LS**.*

Example:

1. Listing all files of the current directory:

   Arguments: **{"token": TokenNumber, "msg_id":1282}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "2013-01-01 11:58:02"} , {"AMBA0002.MP4" : "2013-01-01 11:58:02"} , { "AMBA0003.MP4" : "2013-01-01 11:58:01"} , {"Folder1/" : "2013-01-01 11:58:02"}]}**
   Arguments: **{"token": TokenNumber, "msg_id": 1282, "param":"-D"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "2013-01-01 11:58:02"} , {"AMBA0002.MP4" : "2013-01-01 11:58:02"} , { "AMBA0003.MP4" : "2013-01-01 11:58:01"} , {"Folder1/" :"2013-01-01 11:58:02"}]}**
   Arguments: **{"token": TokenNumber, "msg_id": 1282, "param":"-S"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "4000 bytes"} , {"AMBA0002.MP4" : "8996 bytes"} , { "AMBA0003.MP4" : "4876 bytes"} ,{"Folder1/" :"87654 bytes"}]}**
   Arguments: **{"token": TokenNumber, "msg_id": 1282, "param":"-D -S"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "4000 bytes|2013-01-01 11:58:02"} , {"AMBA0002.MP4" : "8996 bytes|2013-01-01 11:58:02"} , { "AMBA0003.MP4" : "4876 bytes|2013-01-01 11:58:01"} , {"Folder1/" :"87654 bytes|2013-01-01 11:58:02"}]}**

2. Listing all files of the specified directory:

   Arguments: **{"token": TokenNumber, "msg_id":1282, "param":"/tmp/fuse_d/DCIM/100MEDIA/"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "2013-01-01 11:58:02"} , {"AMBA0002.MP4" : "2013-01-01 11:58:02"} , { "AMBA0003.MP4" : "2013-01-01 11:58:01"} , {"Folder1/" : "2013-01-01 11:58:02"}]}**
   Arguments: **{"token": TokenNumber, "msg_id": 1282, "param":"/tmp/fuse_d/DCIM/100MEDIA/ -D"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "2013-01-01 11:58:02"} , {"AMBA0002.MP4" : "2013-01-01 11:58:02"} , { "AMBA0003.MP4" : "2013-01-01 11:58:01"} , {"Folder1/" :"2013-01-01 11:58:02"}]}**
   Arguments: **{"token": TokenNumber, "msg_id": 1282, "param":"/tmp/fuse_d/DCIM/100MEDIA/ -S"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "4000 bytes"} , {"AMBA0002.MP4" : "8996 bytes"} , { "AMBA0003.MP4" : "4876 bytes"} ,{"Folder1/" :"87654 bytes"}]}**
   Arguments: **{"token": TokenNumber, "msg_id": 1282, "param":"/tmp/fuse_d/DCIM/100MEDIA/ -D -S"}**
   Successful return: **{"rval": 0, "msg_id":1282, "listing": [{"AMBA0001.MP4" : "4000 bytes|2013-01-01 11:58:02"} , {"AMBA0002.MP4" : "8996 bytes|2013-01-01 11:58:02"} , { "AMBA0003.MP4" : "4876 bytes|2013-01-01 11:58:01"} , {"Folder1/" :"87654 bytes|2013-01-01 11:58:02"}]}**

   Note: If it is a directory, **AMBA_LS** will add a "/" symbol at the end of the folder name in the listing array.

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.3 AMBA_CD

Message ID:

AMBA_CD msg_id: 0x00000503(1283)

API Overview:

- Type: Function.
- Description: This API changes the current working directory. This command accepts single directory movement.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1283	Remote command number (values must be decimal)
param	destination directory	Destination directory paths

*Table 7-5. Arguments for File System Command API **AMBA_CD**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1283	The **rval** is the response to this remote command.
pwd	current directory	Print Working Directory

*Table 7-6. Returns for File System Command API **AMBA_CD**.*

Example:

If the current path is a:\DCIM\Photo:

Arguments: **{"token": TokenNumber, "msg_id":1283, "param": ".."}**

Successful return: **{"rval": 0, "msg_id":1283, "pwd": "/tmp/fuse_d/DCIM"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.4 AMBA_PWD

Message ID:

AMBA_PWD msg_id: 0x00000504(1284)

API Overview:

- Type: Function.
- Description: This API obtains the current directory.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1284	Remote command number (values must be decimal)

*Table 7-7. Arguments for File System Command API **AMBA_PWD**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1284	The **rval** is the response to this remote command.
pwd	current directory	Print Working Directory

*Table 7-8. Returns for File System Command API **AMBA_PWD**.*

Example:

If the current path is a:\DCIM\Photo:
Arguments: **{"token": TokenNumber, "msg_id":1284}**
Successful return: **{"rval": 0, "msg_id":1284, "pwd": "/tmp/fuse_d/DCIM"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.5 AMBA_GET_FILE

Message ID:

AMBA_GET_FILE msg_id: 0x00000505(1285)

API Overview:

- Type: Function.
- Description: This API is used to retrieve a file. This command differs from a generic file get, such as an HTTPGET, as it has no TCP/IP dependency and works equally well over Bluetooth or WiFi.

  This command is executed in a two-step fashion. First, the camera begins transmitting the requested file (The assumption is that the requested file is found). The return value, **rval**, only implies a successful file retrieval and commencement of transmission to the handheld.

  Upon the completion of file transfer, the camera will send a completion notification **AMBA_NOTIFICATION > GET_FILE_COMPLETE** (Section 7.2.6). The handheld should only infer a successful transfer when it receives this notification.
- Note:

  Send **AMBA_SET_CLNT_INFO** command to specify client information before **AMBA_GET_FILE** starts. Please refer to "Appendix 2 AMBA_GET_FILE" for details on how to implement the client side of the file transfer.

- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1285	Remote command number (values must be decimal)
param	FileName	File of file name. This can be a file in the current directory or a fully qualified path and filename.
offset	Offset_value	Offset value (bytes): 0: For the new get file >0: If it is continuing from a previously disrupted transfer
fetch_size	<size to fetch in bytes>	Size to fetch if this field is specified. Zero or absent implies getting the total file.

*Table 7-9. Arguments for File System Command API **AMBA_GET_FILE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1285	The **rval** is the response to this remote command.
rem_size	<size of file remainder>	Size of requested file in bytes. Full size minus offset (i.e., full file size if the offset is zero).
size	<size of file>	Size of the file

*Table 7-10. Returns for File System Command API **AMBA_GET_FILE**.*

Example:

Arguments: **{"fetch_size" : 0,"msg_id" : 1285,"offset" : 0,"param" : "/tmp/SD0/DCIM/100MEDIA/01020629_0118.MP4","token" : 1}**
Successful return: **{"rval":0,"msg_id":1285,"rem_size":104923136,"size":104923136}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.6 AMBA_CANCEL_FILE_XFER

Message ID:

AMBA_CANCEL_FILE_XFER msg_id: 0x00000507(1287)

API Overview:

- Type: Function
- Description: This API is used to cancel file transfer (both **AMBA_GET_FILE** and **AMBA_PUT_FILE**) between camera and handheld. This command can be employed by the handheld to terminate a file retrieval/upload in the middle of the transfer.
- Direction: From the handheld to the Camera.

- NOTE: There might be no Notification for **AMBA_GET_FILE** or **AMBA_PUT_FILE** sent from camera after the handheld issues this command. If the handheld gets these notifications after issuing this command, it should ignore them.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1287	Remote command number (values must be decimal)
param	FileName	File of file name. It can be a file in the current directory or a fully qualified path and filename.
sent_size (AMBA_PUT_FILE)	<size of data sent> (in bytes)	Size of file (**AMBA_PUT_FILE**) data portion already sent by handheld prior to **AMBA_CANCEL_FILE_XFER** (This field should only exist while canceling **AMBA_PUT_FILE**)

*Table 7-11. Arguments for File System Command API **AMBA_CANCEL_FILE_XFER**.*

Returns:

Key	Value	Description
rval	< 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1287	The **rval** is the response to this remote command.
transferred_size (AMBA_GET_FILE)	<size of data sent> (in bytes)	Size of file (**AMBA_GET_FILE**) data portion already sent by camera prior to **AMBA_CANCEL_FILE_XFER**.

*Table 7-12. Returns for File System Command API **AMBA_CANCEL_FILE_XFER**.*

Example:

To cancel **AMBA_GET_FILE**:
Arguments: **{"token": TokenNumber, "msg_id": 1287, "param": "AMBA0001.MP4"}**
Successful return: **{"rval": 0, "msg_id": 1287, "transferred_size":470000}**

To cancel **AMBA_PUT_FILE**:
Arguments: **{"token":TokenNumber,"msg_id": 1287,"param":"AMBA0001.MP4", "sent_size":342150}**
Successful return: **{"rval":0,"msg_id":1287}**

Error return: **{ "rval": -13, "msg_id":1287} //** The indicated file does not match the transferring file
{ "rval": - 14, "msg_id":1287} // Camera is not transferring file
{ "rval": -26, "msg_id":1287} // Illegal filepath

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.7 AMBA_NOTIFICATION > GET_FILE_COMPLETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x0000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld the latest requested file has been sent.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	7	Remote command number (values must be decimal)
type	"get_file_complete"	Notification of successful transmission
param	<bytes sent> <md5sum>	Number of bytes sent since last **AMBA_GET_FILE** **md5sum** of the requested buffer.

*Table 7-13. Arguments for File System Command Notification **AMBA_NOTIFICATION > GET_FILE_COMPLETE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to verify the number of bytes sent to make sure all the intermittent transfers (if there were any) add up to the total size as well as verify the **md5sum** of the entire requested file.

Example:

Arguments: **{"token": TokenNumber, "msg_id":7, "type":" get_file_complete", "param": [{"bytes sent":30000}, {"md5sum": "4ea92dcc94c42aab5ffe30e20a9ddc91"}]}**

- Chapter 4 "Session Controls"

7.2.8 AMBA_PUT_FILE

Message ID:

AMBA_PUT_FILE msg_id: 0x00000506(1286)

API Overview:

- Type: Function.
- Description:
  - This API is used to place a file in the camera. This command differs from a generic file put, such as an HTTPPUT, as it has no TCP/IP dependency and works equally well over Bluetooth or WiFi.
  - This command is executed in a two-step fashion. First, the camera begins receiving the specified file, assuming that (1) there is sufficient free space, and (2) the handheld has proper write permission at the destination directory. The return value, **rval**, only implies the successful commencement of reception.
  - The **PUT_FILE** command will have three possible outcomes (1) Success, (2) Out of Camera Storage Space, or (3) Interrupted mid-transmission and being timed out. For outcome 1, upon the completion of file reception, as the second step, the camera will send a completion notification **AMBA_NOTIFICATION** > **PUT_FILE_COMPLETE** (Section 7.2.8). The smartphone should only infer a successful transfer when it receives this notification. For outcome 3, the **param** <bytes received> will denote the portion of the file successfully received and the **offset** argument can be used to continue the transmission of the **AMBA_PUT_FILE** from the last successful location.
  - Note:

    Send **AMBA_SET_CLNT_INFO** command to specify client information before **AMBA_PUT_FILE** starts.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1286	Remote command number (values must be decimal)
param	FileName	File of file name. This could be a file in the current directory or a fully qualified path and filename.
size	<FileSize>	Size of the file
md5sum	<md5sum of file>	The md5sum of the requested buffer
offset	Offset_value	Offset value (bytes): 0: New file or overwritten old file with same name >0: Will continue to receive only if offset value matches the amount already received.

*Table 7-14. Arguments for File System Command API **AMBA_PUT_FILE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1286	The **rval** is the response to this remote command.

*Table 7-15. Returns for File System Command API **AMBA_PUT_FILE**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 1286, "param": "file.txt", "size":<size of file. txt>,"md5sum": "4ea92dcc94c42aab5ffe30e20a9ddc91", "offset":0}**
Successful return: **{"rval": 0, "msg_id": 1286}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

7.2.9 AMBA_NOTIFICATION > PUT_FILE_COMPLETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x0000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the latest file requested has been received.
- Direction: From the Camera to the handheld

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	7	Remote command number (values must be decimal)
type	<put file status>	"put_file_complete" - Notification of successful transmission. "timed_out" – Incomplete and **param** <bytes received> denotes the location of the last successful portion. "no_space" – Camera is full.
param	<bytes received> <md5sum>	Number of bytes received since last **AMBA_PUT_FILE**. The md5sum of the complete file that is received

*Table 7-16. Arguments for System Command Notification **AMBA_NOTIFICATION > PUT_FILE_COMPLETE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to verify the **md5sum** of the entire requested file.

Example:

Arguments: **{"token": TokenNumber, "msg_id":7, "type":" put_file_complete", "param": [{"bytes received":30000}, {"md5sum": "4ea92dcc94c42aab5ffe30e20a9ddc91"}**

- Section 4.2.1 "AMBA_START_SESSION"

7.2.10 AMBA_MKDIR

Message ID:

AMBA_MKDIR msg_id: 0x00000509(1289)

API Overview:

- Type: Function.
- Description:
  - This API is used to create the directory at the specified path.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1289	Remote command number (values must be decimal)
param	Directory to be created	Destination directory path. It can be a folder in the current directory or a fully qualified path.

*Table 7-17. Arguments for File System Command API **AMBA_MKDIR**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1289	The **rval** is the response to this remote command.

*Table 7-18. Returns for File System Command API **AMBA_MKDIR**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id":1289, "param": "101MEDIA"}**
Successful return: **{"rval": 0, "msg_id": 1289}**

{"rval": -25, "msg_id": 770} // The requested path is illegal
{"rval": -27, "msg_id": 770} // Directory exists
{"rval": -28, "msg_id": 770} // Cannot create directory at specified path (permission/readonly)
{"rval": -23, "msg_id": 770} // Device does not support this operation

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

8 WiFi Commands

8.1 WiFi Commands: Overview

This section lists the API interface to restart the WiFi service.

8.2 WiFi Commands: List

This section lists the API File System Commands 0x000006XX.

- AMBA_WIFI_RESTART
- AMBA_SET_WIFI_SETTING
- AMBA_GET_WIFI_SETTING
- AMBA_WIFI_STOP
- AMBA_WIFI_START
- AMBA_GET_WIFI_STATUS

8.2.1 AMBA_WIFI_RESTART

Message ID:

AMBA_WIFI_RESTART msg_id: 0x00000601(1537)

API Overview:

- Type: Function.
- Description: This API restarts the WiFi subsystem. This command should be used when WiFi configuration changes are to be applied.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1537	Remote command number (values must be decimal)

*Table 8-1. Arguments for WiFi Command API **AMBA_WIFI_RESTART**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1537	The **rval** is the response to this remote command.

*Table 8-2. Returns for WiFi Command API **AMBA_WIFI_RESTART**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id":1537}**
Successful return: **{"rval": 0, "msg_id":1537}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

8.2.2 AMBA_SET_WIFI_SETTING

Message ID:

AMBA_SET_WIFI_SETTING msg_id: 0x00000602(1538)

API Overview:

- Type: Function.
- Description: This API is used to modify the configuration file of WIFI. Please note that the user should restart the WIFI if one hopes the modified setting is applied.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1538	Remote command number (values must be decimal)
Param	The setting the user wants to modified	The "\n" is the identified symbol if the user wants to modify more than one settings. Please note that the backslash ("\") may be need if the user wants to use any special (pre-saved) symbol.

*Table 8-3. Arguments for WiFi Command API **AMBA_SET_WIFI_SETTING**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1538	The **rval** is the response to this remote command.

*Table 8-4. Returns for WiFi Command API **AMBA_SET_WIFI_SETTING**.*

Example:

Arguments: **{"msg_id": 1538,"param": "AP_SSID=AMBA\nLOCAL_IP=192.128.42.2\nESSID=amba_TEST\nAP_CHANNEL=4","token": TokenNumber}**
Successful return: **{"rval": 0, "msg_id":1538}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

8.2.3 AMBA_GET_WIFI_SETTING

Message ID:

AMBA_GET_WIFI_SETTING msg_id: 0x00000602(1539)

API Overview:

- Type: Function.
- Description: This API is used to get the content of the WIFI configuration.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1539	Remote command number (values must be decimal)

*Table 8-5. Arguments for WiFi Command API **AMBA_GET_WIFI_SETTING**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1539	The **rval** is the response to this remote command.
param	WIFI configuration	The complete content of the WIFI configuration. The "\n" is the identified symbol between two settings.

*Table 8-6. Returns for WiFi Command API **AMBA_GET_WIFI_SETTING**.*

Example:

Arguments: **{"msg_id" : 1539,"token" : TokenNumber}**
Successful return: **{"rval":0,"msg_id":1539,"param":"WIFI_MODE=ap\nESSID=amba_boss\nPASS-WORD=1234567890\nSTA_DEVICE_NAME=amba1\nSTA_SKIP_SCAN=yes\nP2P_AUTO_CONNECT=no\nP2P_CONNECT_PREFIX=amba\nAP_SSID=amba_boss\nLOCAL_IP=192.168.42.1\nLOCAL_NET-MASK=255.255.255.0\nDHCP_IP_START=192.168.42.2\nDHCP_IP_END=192.168.42.6\nAP_CHANNEL=0\nAP_MAXSTA=5\nAP_PUBLIC=no\nAP_PASSWD=1234567890\n"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

8.2.4 AMBA_WIFI_STOP

Message ID:

AMBA_WIFI_STOP msg_id: 0x00000603(1540)

API Overview:

- Type: Function.
- Description: This API is used to stop the WIFI.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1540	Remote command number (values must be decimal)

*Table 8-7. Arguments for WiFi Command API **AMBA_WIFI_STOP**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1540	The **rval** is the response to this remote command.

*Table 8-8. Returns for WiFi Command API **AMBA_WIFI_STOP**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id":1540}**
Successful return: **{"rval": 0, "msg_id":1540}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

8.2.5 AMBA_WIFI_START

Message ID:

AMBA_WIFI_START msg_id: 0x00000604(1541)

API Overview:

- Type: Function.
- Description: This API is used to start the WIFI.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1541	Remote command number (values must be decimal)

*Table 8-9. Arguments for WiFi Command API **AMBA_WIFI_START**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1541	The **rval** is the response to this remote command.

*Table 8-10. Returns for WiFi Command API **AMBA_WIFI_START**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id":1541}**
Successful return: **{"rval": 0, "msg_id":1541}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

8.2.6 AMBA_GET_WIFI_STATUS

Message ID:

AMBA_GET_WIFI_STATUS msg_id: 0x00000605(1542)

API Overview:

- Type: Function.
- Description: This API is used to get the current status of WIFI.
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1542	Remote command number (values must be decima)

*Table 8-11. Arguments for WiFi Command API **AMBA_GET_WIFI_STATUS**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1542	The **rval** is the response to this remote command.
Param	WIFI Status	The current status of WIFI

*Table 8-12. Returns for WiFi Command API **AMBA_GET_WIFI_STATUS**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id":1542}**
Successful return: **{"rval":0,"msg_id":1542,"param":{"status":"enabled", "mode":"ap","SSID":"amba_boss","IP":"192.168.42.1","MAC":"00:03:7F:FA:B3:4D","CONN_ amount":"1","CONN_list":["ac:22:0b:92:36:14"]}}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

9 Media Commands

9.1 Media Commands: Overview

This chapter covers the API interface used to interact with media (photo and video) files.

9.2 Media Commands: List

This section lists the API Media Commands 0x000004XX.

- AMBA_GET_THUMB
- AMBA_GET_MEDIAINFO
- AMBA_SET_MEDIA_ATTRIBUTE

9.2.1 AMBA_GET_THUMB

Message ID:

AMBA_GET_THUMB msg_id: 0x00000401(1025)

API Overview:

- Type: Function.
- Description: This API can be used to retrieve a thumbnail file. After the handheld sends this API, it should receive thumbnail data from the data port. The returned result will be sent to the handheld after the thumbnail is transmitted completely.
- Direction: From the handheld to the camera
- Note: There will be no notification sent for this API in comparsion to **AMBA_GET_FILE**. Please use the md5sum field in the API response to check the correctness of the received thumbnail data.
- Note: Send **AMBA_SET_CLNT_INFO** command to specify client information before **AMBA_GET_THUMB** starts.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1025	Remote command number (values must be decimal)
param	FileName	File containing the file name. It can be a file in the current directory or a fully qualified path and file name.
type	<Accept Thumb_Type>	Thumb: Only thumbnail IDR: Only IDR Fullview: Full-resolution only

*Table 9-1. Arguments for Media Command API **AMBA_GET_THUMB**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1025	The **rval** is the response to this remote command.
thumb_file	<Name of Thumbnail>	Name of the associated thumbnail filename. Field will be absent if the associated thumbnail is missing.
size	<Size of File>	Size of the thumbnail file
type	<Return Thumb_Type>	Thumb: Only thumbnail IDR: Only IDR Fullview: Full-resolution only
md5sum	<md5sum of Image>	The **md5sum** of the requested file

*Table 9-2. Returns for Media Command API **AMBA_GET_THUMB**.*

Example:

Arguments: **{"token": TokenNumber, "msg_id": 1025, "param": "AMBA0001.MP4", "type":"thumb" }**
Successful return: **{"rval": 0, "msg_id": 1025, "size":<size of AMBA0001. MP4>,"type":"thumb", "md5sum": "4ea92dcc94c42aab5ffe30e20a9ddc91"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION""
- Chapter 13 "System Errors"

9.2.2 AMBA_GET_MEDIAINFO

Message ID:

AMBA_GET_MEDIAINFO msg_id: 0x00000402(1026)

API Overview:

- Type: Function.
- Description: This API can be used to retrieve information about a specific photo or video. The information list includes the thumbnail filename, size, date (creation/modification), resolution, duration (if video) and media type (**mov** – video, **img** – photo, **thm** - thumbnail).
- Direction: From the handheld to the Camera.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1
msg_id	1026	Remote command number (values must be decimal)
type (optional)	“msec”, “sec”	This field is optional. It indicates the unit of the returned duration as in mini seconds or seconds. The duration will be in seconds if this field is absent.
param	Filename	The specified file

*Table 9-3. Arguments for Media Command API **AMBA_GET_MEDIAINFO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1026	The **rval** is the response to this remote command.
thumb_file (Option)	<name of thumbnail>	Name of the associated thumbnail filename. The field will be absent if the associated thumbnail is missing.
size	<size of file>	Size of **FileName**
date	<date / time>	Time and date of the file creat/last modification in YYYY-MM-DD HH:MM:SS
resolution	<resolution>	Dimension of the file in strings “basexheight”
duration	<duration if video>	Duration of this video file. By default, this value is represented in seconds. If “type”=”msec” is carried within the command, this value is represented in mini-seconds. Field will be absent if **img** is specified.
media_type	<type of media>	The type of media, **mov/img/thm**

*Table 9-4. Returns for Media Command API **AMBA_GET_MEDIAINFO**.*

Example:

Arguments: **: {"token": TokenNumber, "msg_id": 1026, "param": "AMBA0001.MP4" }**
Successful return: **{"rval": 0, "msg_id": 1026, "thumb_file": "AMBA0001. THM", "size":<size of AMBA0001. MP4>, "date": "2012-12-31 23:23:59", "resolution": "1920x1080", "duration":"600", "media_type": "mov"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

9.2.3 AMBA_SET_MEDIA_ATTRIBUTE

Message ID:

AMBA_SET_MEDIA_ATTRIBUTE msg_id: 0x00000403(1027)

API Overview:

- Type: Function
- Description: This API sets the read/write attributes of the specified media file. This command is used to protect the media file from being deleted by the circle-recording mechanism. This API is specifically for media objects managed via DCF.
- Direction: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by the **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1027	Remote command number (values must be decimal)
type	attributes	Specify the attribute to set the set:: 0x0000: Writable 0x0001: Read only
param	Filename	File of file name. It can be a file in the current directory or a fully qualified path and file name.

*Table 9-5. Arguments for Media Command API **AMBA_SET_MEDIA_ATTRIBUTE**.*

Returns:

Key	Value	Description
rval	< 0	0: Success <0: Error as defined in Chapter 13.
msg_id	1027	The **rval** is the response to this remote command.

*Table 9-6. Returns for Media Command API **AMBA_SET_MEDIA_ATTRIBUTE**.*

Example:

Arguments: **: {"token": TokenNumber, "msg_id": 1027, "type":1,"param": "/tmp/fuse_d/DCIM/100MEDIA/AMBA_0001.mp4"} //set as read-only**
Successful return: **{"rval": 0, "msg_id": 1027}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"
- Chapter 13 "System Errors"

10 Query Commands

10.1 Query Commands: Overview

This section discusses the API query commands sent from the camera to the handheld device.

10.2 Query Commands: List

This section lists the API File Query Commands 0x000007XX.

- AMBA_QUERY_SESSION_HOLDER

10.2.1 AMBA_QUERY_SESSION_HOLDER

Message ID:

AMBA_QUERY_SESSION_HOLDER msg_id: 0x00000701(1793)

API Overview:

- Type: Function
- Description: The camera will send out this query command to the client who currently holds the session (i.e., has locked the **token**) in the event that another client tries to create new session. When this command is received, a timeout period begins. Within this period, the client who holds the session is obligated to respond with its **token** in order to indicate that the session is still alive. Otherwise, the camera will close the existing session and allow the creation of a new session once the timeout period has elapsed. The default timeout period is 800 ms.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	1793	Remote command number (values must be decimal)

*Table 10-1. Arguments for Query Command API **AMBA_QUERY_SESSION_HOLDER**.*

Returns:

Key	Value	Description
rval	0, - 1	= 0 = - 1: Session holder claim gives up this session.
msg_id	1793	Remote command number (values must be decimal)
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)

*Table 10-2. Returns for Query Command API **AMBA_QUERY_SESSION_HOLDER**.*

Example:

Arguments: **{"msg_id":1793}**
Successful return: **{"rval": 0,"msg_id":1793 ,"token": TokenNumber}**
Error return: {"rval":-1, "msg_id": 1793,"token": 0}
Timeout, or error token number

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11 Notifications

11.1 Notifications: Overview

Notifications proceed in the opposite direction with respect to other typical commands. Instead of the camera receiving command requests from the handheld, the camera sends notifications to the handheld regarding changes in camera status.

Notifications are a means of conveying camera events to the handheld. Camera events of interest such as manual operation by the end user (e.g., pressing the record button) trigger the sending of notifications. Certain camera-related events, such as the camera powering off or an HDMI connection, will result in a notification indicating that the handheld should discontinue the current session.

Notifications are also listed throughout this document as command responses. For example, the camera will send a notification when its firmware upgrade is complete.

11.2 Notifications: List

- AMBA_NOTIFICATION > DISCONNECT_HDMI
- AMBA_NOTIFICATION > DISCONNECT_SHUTDOWN
- AMBA_NOTIFICATION > STARTING_VIDEO_RECORD
- AMBA_NOTIFICATION > VIDEO_RECORD_COMPLETE
- AMBA_NOTIFICATION > PHOTO_TAKEN
- AMBA_NOTIFICATION > CONTINUE_CAPTURE_START
- AMBA_NOTIFICATION > CONTINUE_CAPTURE_STOP
- AMBA_NOTIFICATION > CONTINUE_BURST_START
- AMBA_NOTIFICATION > CONTINUE_BURST_COMPLETE
- AMBA_NOTIFICATION > LOW_BATTERY_WARNING
- AMBA_NOTIFICATION > LOW_STORAGE_WARNING
- AMBA_NOTIFICATION > TIMELAPSE_VIDEO_STATUS
- AMBA_NOTIFICATION > TIMELAPSE_PHOTO_STATUS
- AMBA_NOTIFICATION > CAMERA_CONNECT_TO_PC
- AMBA_NOTIFICATION > LOG_UPDATED
- AMBA_NOTIFICATION > POWER_MODE_CHANGE
- AMBA_NOTIFICATION > VF_START
- AMBA_NOTIFICATION > VF_STOP
- AMBA_NOTIFICATION > AUTO_FILE_DELETE
- AMBA_NOTIFICATION > STORAGE_RUNOUT
- AMBA_NOTIFICATION > STORAGE_IO_ERROR
- AMBA_NOTIFICATION > LOW_SPEED_CARD
- AMBA_NOTIFICATION > MUXER_INDEX_LIMIT
- AMBA_NOTIFICATION > MUXER_FILE_SIZE_LIMIT

- AMBA_NOTIFICATION > CARD_REMOVED
- AMBA_NOTIFICATION > CARD_INSERTED
- AMBA_NOTIFICATION > CANNOT_ISSUE_PIV
- AMBA_NOTIFICATION > START_EVENT_RECORD
- AMBA_NOTIFICATION > EVENT_RECORD_COMPLETE
- AMBA_NOTIFICATION > FORCE_DISCONNECT
- AMBA_NOTIFICATION > MENU_ON
- AMBA_NOTIFICATION > MENU_OFF
- AMBA_NOTIFICATION > THUMB_MOTION_ON
- AMBA_NOTIFICATION > THUMB_MOTION_OFF

11.2.1 AMBA_NOTIFICATION > DISCONNECT_HDMI

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by cameras that cannot operate as a camera while simultaneously using HDMI. This notification signals to the handheld that the current session has been terminated by the camera because a HDMI cable has been attached, and the camera is now driving a display (TV or monitor) via the HDMI interface.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"disconnect_HDMI"	Notification type: Disconnect for HDMI operation.

*Table 11-1. Arguments for Notification **AMBA_NOTIFICATION > DISCONNECT_HDMI**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change in the camera mode to HDMI.

Example:

Arguments: **{"msg_id": 7,"type":"disconnect_HDMI"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.2 AMBA_NOTIFICATION > DISCONNECT_SHUTDOWN

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the session has been terminated by the camera because it is about to shutdown.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	“disconnect_shutdown”	Notification type: Disconnect because of shutdown.

*Table 11-2. Arguments for Notification **AMBA_NOTIFICATION > DISCONNECT_SHUTDOWN**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the session termination.

Example:

Arguments: **{“msg_id”: 7, “type”:”disconnect_shutdown”}**

See Also:

- Section 4.2.1 “AMBA_START_SESSION”

11.2.3 AMBA_NOTIFICATION > STARTING_VIDEO_RECORD

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the record button has been activated manually and the recording footage has commenced.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	“starting_video_record”	Notification type: A video recording has started.

*Table 11-3. Arguments for Notification **AMBA_NOTIFICATION > STARTING_VIDEO_RECORD**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the start of video recording.

Example:

Arguments: **{“msg_id”: 7,“type”:”starting_video_record”}**

See Also:

- Section 4.2.1 “AMBA_START_SESSION”

11.2.4 AMBA_NOTIFICATION > VIDEO_RECORD_COMPLETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the record button has been activated manually to stop video recording. A new video file has been created as the result of pressing the button.
- Direction: From the Camera to the handheld

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"video_record_complete"	Notification type: A recording has stopped and a new video file has been created and stored.
param	<FileName>	Return parameter is the full path of the video recorded (e.g. `/tmp/fuse_d/DCIM/100MEDIA/AMBA0031.MP4`)

*Table 11-4. Arguments for Notification **AMBA_NOTIFICATION > VIDEO_RECORD_COMPLETE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect that the video recording has been terminated by pressing the manual record button.

Example:

Arguments: **{"msg_id": 7, "type":"video_record_complete", "param":"/tmp/fuse_d/DCIM/100MEDIA/AMBA0031.MP4"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.5 AMBA_NOTIFICATION > PHOTO_TAKEN

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the shutter button has been manually pressed. A new photo file has been created as the result of pressing the button.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"photo_taken"	Notification type: A photo has been taken and a new photo file has been created and stored.
param	<saved file name>	File name including the fully qualified path.

*Table 11-5. Arguments for Notification **AMBA_NOTIFICATION > PHOTO_TAKEN**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect that a photo has been manually taken by the camera.

Example:

Arguments: **{"msg_id": 7, "type":"photo_taken", "param":"/tmp/fuse_d/DCIM/100MEDIA/AMBA0031.JPG"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.6 AMBA_NOTIFICATION > CONTINUE_CAPTURE_START

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the shutter button has been manually pressed and the precise quality continue capture operation has started.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"continue_capture_start"	Notification type: Precise quality continue capture operation has started.

*Table 11-6. Arguments for Notification **AMBA_NOTIFICATION > CONTINUE_CAPTURE_START**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect that the precise continue capture has started.

Example:

Arguments: **{"msg_id": 7, "type":"continue_capture_start"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.7 AMBA_NOTIFICATION > CONTINUE_CAPTURE_STOP

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the shutter button has been manually pressed and that the precise quality continue capture operation has stopped.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"continue_capture_stop"	Notification type: Precise quality continue capture operation has stopped.
param	<saved file name>	File name including fully qualified path. (last captured file)

*Table 11-7. Arguments for Notification **AMBA_NOTIFICATION > CONTINUE_CAPTURE_STOP**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect that the precise continue capture has stopped.

Example:

Arguments: **{"msg_id": 7, "type":"continue_capture_stop"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.8 AMBA_NOTIFICATION > CONTINUE_BURST_START

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the shutter button has been manually pressed and the burst continue capture operation has been started.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"continue_burst_ start"	Notification type: Burst continue capture operation has started.
param	<saved file name>	File name including fully qualified path. (last captured file)

*Table 11-8. Arguments for Notification **AMBA_NOTIFICATION > CONTINUE_BURST_START**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect that the burst continue capture has started.

Example:

Arguments: **{"msg_id": 7, "type":"continue_burst_start"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.9 AMBA_NOTIFICATION > CONTINUE_BURST_COMPLETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the shutter button has been manually pressed and that the burst continue capture operation has been completed.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"continue_burst_ complete"	Notification type: Burst continue capture operation has stopped.
param	<saved file name>	File name including fully qualified path. (last captured file)

*Table 11-9. Arguments for Notification **AMBA_NOTIFICATION > CONTINUE_BURST_COMPLETE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect that the burst continue capture has completed.

Example:

Arguments: **{"msg_id": 7, "type":"continue_burst_complete"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.10 AMBA_NOTIFICATION > LOW_BATTERY_WARNING

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the camera's battery is running low.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"low_battery_warning"	Triggering threshold battery charge percentage can be set as part of the camera's configuration
param	RemainingPercentage	RemainingPercentage represents the percent of the remaining battery charge.

*Table 11-10. Arguments for Notification **AMBA_NOTIFICATION >LOW_BATTERY_WARNING**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the low battery charge status.

Example:

Arguments: **{"msg_id": 7, "type":"low_battery_warning", "param":10}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.11 AMBA_NOTIFICATION > LOW_STORAGE_WARNING

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the camera is running low on storage space.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"low_storage_warning"	Triggering threshold remaining storage space percentage can be set as part of the camera's configuration.
param	RemainingPercentage	The remaining storage space as a percentage of the initial amount of space.

*Table 11-11. Arguments for Notification **AMBA_NOTIFICATION > LOW_STORAGE_WARNING**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the low storage availability status.

Example:

Arguments: **{"msg_id": 7, "type":"low_storage_warning", "param":10}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.12 AMBA_NOTIFICATION > TIMELAPSE_VIDEO_STATUS

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the setting of the timelapse photo was manually changed.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"timelapse_video_status"	Notification of the timelapse_video status change.
param	New setting	Setting of the timelapse_video

*Table 11-12. Arguments for Notification **AMBA_NOTIFICATION > TIMELAPSE_VIDEO_STATUS**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the manual change in the timelapse video setting.

Example:

Arguments: **{"msg_id": 7, "type":"timelapse_video_status", "param": "off"}**
Arguments: **{"msg_id": 7, "type":"timelapse_video_status", "param": "2"} //2 seconds**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.13 AMBA_NOTIFICATION > TIMELAPSE_PHOTO_STATUS

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the setting of the timelapse photo was manually changed.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"timelapse_photo_status"	Notification of the timelapse_video status change.
param	New setting	Setting of the timelapse_photo

*Table 11-13. Arguments for Notification **AMBA_NOTIFICATION > TIMELAPSE_PHOTO_STATUS**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the timelapse photo status (off or on).

Example:

Arguments: **{"msg_id": 7, "type":"timelapse_photo_status", "param": "off"}**
Arguments: **{"msg_id": 7, "type":"timelapse_photo_status", "param": "2"} //2 seconds**

See Also:

- Chapter 4 "Session Controls"

11.2.14 AMBA_NOTIFICATION > CAMERA_CONNECT_TO_PC

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the camera is connected to a PC. In this state, the camera's storage is accessed by the computer via USB Mass Storage Class. The camera will stop video recording and/or stop time-lapse photography. The camera will close connection to the handheld after this notification is sent.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"camera_connect_to_pc"	Camera is plugged in to the computer via the USB cable.
param	None	This parameter is not applicable.

*Table 11-14. Arguments for Notification **AMBA_NOTIFICATION > CAMERA_CONNECT_TO_PC**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to respond to the camera severing connection while connected to PC via the USB cable.

Example:

Arguments: **{"msg_id": 7, "type":"camera_connect_to_pc"}**

See Also:

- Chapter 4 "Session Controls"

11.2.15 AMBA_NOTIFICATION > LOG_UPDATED

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that there is a file available to be uploaded to a specified server.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"log_updated"	Notification type: There is a log file to be uploaded to a specified server.
param	<file name>	File name with Full path
server	<server information>	The address for server, such as: "name@server.com": mail address "http://www.server.com": HTTP address "ftp://server.com": FTP address

*Table 11-15. Arguments for Notification **AMBA_NOTIFICATION > LOG_UPDATED**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to retrieve the log file with the **AMBA_GET_FILE** command and then upload it to the specified server.

Example:

Arguments: **{"msg_id": 7, "type":"log_updated", "param":"/tmp/fuse_d/log.xml", "server":"log_svc@ambarella.com"}**

See Also:

- Chapter 4 "Session Controls"

11.2.16 AMBA_NOTIFICATION > POWER_MODE_CHANGE

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that there is a camera power mode change occurring.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"power_mode_change"	Notification type: There is a camera power mode change.
param	<mode>	The power mode change which is made on the camera: "cam_stb": Standby camera "cam_off": Power-off camera

*Table 11-16. Arguments for Notification **AMBA_NOTIFICATION > POWER_MODE_CHANGE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change in the camera's power mode.

Example:

Arguments: **{"msg_id": 7, "type":"power_mode_change", "param":"cam_stb"}**

See Also:

- Chapter 4 "Session Controls"

11.2.17 AMBA_NOTIFICATION > VF_START

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the view finder stream is enabled.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"vf_start"	Notification type: Viewfinder is enabled
param	None	This parameter is not applicable.

*Table 11-17. Arguments for Notification **AMBA_NOTIFICATION > VF_START**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change in the camera's view finder.

Example:

Arguments: **{"msg_id": 7, "type":"vf_start"}**

See Also:

- Chapter 4 "Session Controls"

11.2.18 AMBA_NOTIFICATION > VF_STOP

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the viewfinder stream is disabled.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"vf_stop"	Notification type: View finder is disabled.
param	None	This parameter is not applicable.

*Table 11-18. Arguments for Notification **AMBA_NOTIFICATION > VF_STOP**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change in the camera's viewfinder.

Example:

Arguments: **{"msg_id": 7, "type":"vf_stop"}**

See Also:

- Chapter 4 "Session Controls"

11.2.19 AMBA_NOTIFICATION > AUTO_FILE_DELETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that a file has been automatically deleted by CarDV cyclic-recording mechanism.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	“auto_file_delete”	Notification type: A file has been automatically deleted by CarDV cyclic-recording mechanism.
param	<filename>	The full path of the deleted file

*Table 11-19. Arguments for Notification **AMBA_NOTIFICATION > AUTO_FILE_DELETE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the content of file list change.

Example:

Arguments: **{“msg_id”: 7, “type”:”auto_file_delete”, “param”:”/tmp/fuse_d/DCIM/100MEDIA/filename.mp4”}**

See Also:

- Chapter 4 “Session Controls”

11.2.20 AMBA_NOTIFICATION > STORAGE_RUNOUT

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that the storage of the camera is not enough and that the recording has been stopped.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"STORAGE_RUNOUT"	Notification type: Camera has run out of storage.

*Table 11-20. Arguments for Notification **AMBA_NOTIFICATION > STORAGE_RUNOUT**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"STORAGE_RUNOUT"}**

See Also:

- Chapter 4 "Session Controls"

11.2.21 AMBA_NOTIFICATION > STORAGE_IO_ERROR

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification
- Description: This API is used by the camera to signal to the handheld that there is IO error on the storage of the camera and that recording has been stopped.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"STORAGE_IO_ERROR"	Notification type: There is IO error in the camera storage.

*Table 11-21. Arguments for Notification **AMBA_NOTIFICATION > STORAGE_IO_ERROR**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change of state of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"STORAGE_IO_ERROR"}**

See Also:

- Chapter 4 "Session Controls"

11.2.22 AMBA_NOTIFICATION > LOW_SPEED_CARD

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the speed of the SD card of the camera is too slow (which causes the internal buffer of the camera to be full) and that the recording has been stopped.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"LOW_SPEED_CARD"	Notification type: The speed of the SD card of the camera is too slow.

*Table 11-22. Arguments for Notification **AMBA_NOTIFICATION > LOW_SPEED_CARD**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change of state of the camera.

Example:

Arguments: **{"msg_id": 7, "type":" LOW_SPEED_CARD"}**

See Also:

- Chapter 4 "Session Controls"

11.2.23 AMBA_NOTIFICATION > MUXER_INDEX_LIMIT

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the record has been stopped due to some internal error from the muxer.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"MUXER_INDEX_LIMIT"	Notification type: Internal muxer error causes the record to stop.

*Table 11-23. Arguments for Notification **AMBA_NOTIFICATION > MUXER_INDEX_LIMIT**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change of state of the camera.

Example:

Arguments: **{"msg_id": 7, "type":" MUXER_INDEX_LIMIT"}**

See Also:

- Chapter 4 "Session Controls"

11.2.24 AMBA_NOTIFICATION > MUXER_FILE_SIZE_LIMIT

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the record has been stopped as the file size has reached the limit.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"MUXER_FILE_SIZE_LIMIT"	Notification type: File size reaches the limit.

*Table 11-24. Arguments for Notification **AMBA_NOTIFICATION > MUXER_FILE_SIZE_LIMIT**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change of state of the camera.

Example:

Arguments: **{"msg_id": 7, "type":" MUXER_FILE_SIZE_LIMIT"}**

See Also:

- Chapter 4 "Session Controls"

11.2.25 AMBA_NOTIFICATION > CARD_REMOVED

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the SD card has been removed and that recording has been stopped.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"CARD_REMOVED"	Notification type: SD card removed.

*Table 11-25. Arguments for Notification **AMBA_NOTIFICATION > CARD_REMOVED**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the change of state of the camera.

Example:

Arguments: **{"msg_id": 7, "type:" CARD_REMOVED"}**

See Also:

- Chapter 4 "Session Controls"

11.2.26 AMBA_NOTIFICATION > CARD_INSERTED

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the SD card has been inserted.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"CARD_INSERTED"	Notification type: SD card inserted

*Table 11-26. Arguments for Notification **AMBA_NOTIFICATION > CARD_INSERTED**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"CARD_INSERTED"}**

See Also:

- Section 4.2.1 "AMBA_START_SESSION"

11.2.27 AMBA_NOTIFICATION > CANNOT_ISSUE_PIV

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the execution of the PIV is blocked due to performance issues.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"CANNOT_ISSUE_PIV"	Notification type: Cannot issue PIV.

*Table 11-27. Arguments for Notification **AMBA_NOTIFICATION > CANNOT_ISSUE_PIV**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the failure of the PIV operation.

Example:

Arguments: **{"msg_id": 7, "type:" CANNOT_ISSUE_PIV"}**

See Also:

- Chapter 4 "Session Controls"

11.2.28 AMBA_NOTIFICATION > START_EVENT_RECORD

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the event record is triggered.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"start_event_record"	Notification type: Camera starts event record

*Table 11-28. Arguments for Notification **AMBA_NOTIFICATION > START_EVENT_RECORD**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type:" start_event_record"}**

See Also:

- Chapter 4 "Session Controls"

11.2.29 AMBA_NOTIFICATION > EVENT_RECORD_COMPLETE

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the event record has been completed.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"event_record_complete"	Notification type: Event record completes
param	<recorded clip file path>	Return parameter is the full path of the video recorded (/tmp/fuse_d/EVENT/10161724_0061_E0001.MP4).

*Table 11-29. Arguments for Notification **AMBA_NOTIFICATION > EVENT_RECORD_COMPLETE**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"event_record_complete", "param":"/tmp/fuse_d/EVENT/10161724_0061_E0001.MP4"}**

See Also:

- Chapter 4 "Session Controls"

11.2.30 AMBA_NOTIFICATION > FORCE_DISCONNECT

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the current session is terminated by the camera.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"force_disconnect"	Notification type: Camera has run out of storage.
param	<types of cause>	The reason why camera is sending this notification. This can help handheld to reflect the cause of disconnection to the user.

*Table 11-30. Arguments for Notification **AMBA_NOTIFICATION > FORCE_DISCONNECT**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"force_disconnect","param":"operated by user"}**

See Also:

- Chapter 4 "Session Controls"

11.2.31 AMBA_NOTIFICATION > MENU_ON

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the menu of the camera is turned on by the user.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"force_disconnect"	Notification type: Camera has run out of storage.
param	"MENU_ON"	Notification type: Camera's menu is turned on by the user.

*Table 11-31. Arguments for Notification **AMBA_NOTIFICATION > MENU_ON**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"MENU_ON"}**

See Also:

- Chapter 4 "Session Controls"

11.2.32 AMBA_NOTIFICATION > MENU_OFF

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the menu of the camera is turned off by the user.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"force_disconnect"	Notification type: Camera has run out of storage.
param	"MENU_OFF"	Notification type: Camera's menu is turned off by the user.

*Table 11-32. Arguments for Notification **AMBA_NOTIFICATION > MENU_OFF**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"MENU_OFF"}**

See Also:

- Chapter 4 "Session Controls"

11.2.33 AMBA_NOTIFICATION > THUMB_MOTION_ON

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the thumb motion mode of the camera is turned on by user.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"force_disconnect"	Notification type: Camera has run out of storage.
param	"THUMB_MOTION_ON"	Notification type: Camera's thumb motion mode is turned on by the user.

*Table 11-33. Arguments for Notification **AMBA_NOTIFICATION > THUMB_MOTION_ON**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"THUMB_MOTION_ON"}**

See Also:

- Chapter 4 "Session Controls"

11.2.34 AMBA_NOTIFICATION > THUMB_MOTION_OFF

Message ID:

AMBA_NOTIFICATION msg_id: 0x00000007(7)

API Overview:

- Type: Notification.
- Description: This API is used by the camera to signal to the handheld that the thumb motion mode of the camera is turned off by the user.
- Direction: From the Camera to the handheld.

Arguments:

Key	Value	Description
msg_id	7	Remote command number (values must be decimal)
type	"force_disconnect"	Notification type: Camera has run out of storage.
param	"THUMB_MOTION_OFF"	Notification type: Camera's thumb motion mode is turned off by the user.

*Table 11-34. Arguments for Notification **AMBA_NOTIFICATION > THUMB_MOTION_OFF**.*

Returns:

None. This command is to be received and acted upon by the handheld. It is the responsibility of the handheld to reflect the state change of the camera.

Example:

Arguments: **{"msg_id": 7, "type":"THUMB_MOTION_OFF"}**

See Also:

- Chapter 4 "Session Controls"

12 Keywords

12.1 Keywords: Overview

This section lists keywords used in the get/set setting commands. Each keyword must be uniquely defined to avoid confusion with regards to the parameter in use.

12.2 Keywords: List

- **camera_connect_to_pc** - Notification
- **disconnect_HDMI** - Notification
- **disconnect_shutdown** - Notification
- **fw_upgrade_complete** - Notification
- **low_battery_warning** - Notification
- **low_storage_warning** - Notification
- **photo_taken** - Notification
- **starting_video_record** - Notification
- **timelapse_photo_interval_change** - Notification
- **timelapse_photo_status** - Notification
- **video_record_complete** - Notification

13 System Errors

13.1 System Errors: Overview

This chapter lists the possible return values (**rval**) when errors are encountered.

13.2 System Errors: Error Code List

- UNKNOWN_ERROR = -1
- SESSION_START_FAIL = -3
- INVALID_TOKEN = -4
- REACH_MAX_CLNT = -5
- JSON_PACKAGE_ERROR = -7
- JSON_PACKAGE_TIMEOUT = -8
- JSON_SYNTAX_ERROR = -9
- INVALID_OPTION_VALUE = -13
- INVALID_OPERATION = -14
- HDMI_INSERTED = -16
- NO_MORE_SPACE = -17
- CARD_PROTECTED = -18
- NO_MORE_MEMORY = -19
- PIV_NOT_ALLOWED = -20
- SYSTEM_BUSY = -21
- APP_NOT_READY = -22
- OPERATION_UNSUPPORTED = -23
- INVALID_TYPE = -24
- INVALID_PARAM = -25
- INVALID_PATH = -26
- DIR_EXIST = -27
- PERMISSION_DENIED = -28
- AUTHENTICATION_FAILED = -29

13.2.1 UNKNOWN_ERROR = -1

Error Value:

UNKNOWN_ERROR = -1

Error Description:

- This error is returned for all miscellaneous problems. Other error values are relatively more specific.
- If an attempted command results in a problem, but the actual cause is not known, this error code is returned.

13.2.2 SESSION_START_FAIL = -3

Error Value:

SESSION_START_FAIL = -3

Error Description:

- This is the response to **AMBA_START_SESSION** if session creation fails.

13.2.3 INVALID_TOKEN = -4

Error Value:

INVALID_TOKEN = -4

Error Description:

- This error indicates that the client sends a command with invalid_token, which means that either the client has never successfully acquire a valid token or the client has lost its original session token to another device.

13.2.4 REACH_MAX_CLNT = -5

Error Value:

REACH_MAX_CLNT = -5

Error Description:

- This error indicates that the camera has reached the maximum number of simultaneous client connections. When this error is returned, the camera cannot accept new connection requests.

13.2.5 JSON_PACKAGE_ERROR = -7

Error Value:

JSON_PACKAGE_ERROR = -7

Error Description:

- A JSON command cannot be nested. This error message is sent when two successive left braces, "{ {" are received without a first matching right brace "}".

13.2.6 JSON_PACKAGE_TIMEOUT = -8

Error Value:

JSON_PACKAGE_TIMEOUT = -8

Error Description:

- This error is sent to indicate a timeout on incomplete JSON commands. If a matching closing right brace "}" is not received for an opening left brace "{" within a timeout length of 5 seconds, this error is generated.

13.2.7 JSON_SYNTAX_ERROR = -9

Error Value:

JSON_SYNTAX_ERROR = -9

Error Description:

- This error code is sent in response to JSON commands containing syntax errors. The **key:value** pairing accepts alphanumeric characters.

13.2.8 INVALID_OPTION_VALUE = -13

Error Value:

INVALID_OPTION_VALUE = -13

Error Description:

- If the **AMBA_SET_SETTING** command is received with an invalid option, this error is returned.

13.2.9 INVALID_OPERATION = -14

Error Value:

INVALID_OPERATION = -14

Error Description:

- This error is returned when an invalid or unknown command is received.

13.2.10 HDMI_INSERTED = -16

Error Value:

HDMI_INSERTED = -16

Error Description:

- This error is returned when a handheld attempts to execute the **AMBA_START_SESSION** command while the camera is connected to a HDMI device.

13.2.11 NO_MORE_SPACE = -17

Error Value:

NO_MORE_SPACE = -17

Error Description:

- This error is returned when there no more space is available in the SD card of the camera. There will be no additional photos stored nor movies recorded.

13.2.12 CARD_PROTECTED = -18

Error Value:

CARD_PROTECTED = -18

Error Description:

- The camera SD card currently is locked in read only mode. The camera cannot record movies or snap pictures until the SD card is unlocked.

13.2.13 NO_MORE_MEMORY = -19

Error Value:

NO_MORE_MEMORY = -19

Error Description:

- This error is returned when the latest command causes the camera to exhaust the available memory.

13.2.14 PIV_NOT_ALLOWED = -20

Error Value:

PIV_NOT_ALLOWED = -20

Error Description:

- This error code indicates that the Photo in Video (PIV) operation is not currently permitted. This may be caused by dedicated video resolution settings or by enabling dedicated camera functions.

13.2.15 SYSTEM_BUSY = -21

Error Value:

SYSTEM_BUSY = -21

Error Description:

- This error code is returned when the client issues a request while the camera is not in idle mode. For example, this code will be returned if the client sends a set setting request while the camera is in record mode. Another example is if the client requests to format the SD card during a file transfer operation.

13.2.16 APP_NOT_READY = -22

Error Value:

APP_NOT_READY = -22

Error Description:

- This error code is returned when the camera application is not initialized. For example, this code would be returned if the client requests a command while the camera is switching modes.

13.2.17 OPERATION_UNSUPPORTED = -23

Error Value:

OPERATION_UNSUPPORTED = -23

Error Description:

- This error code is returned when the camera application does not support the operation requested by client as that there is no such use case in the camera application.

13.2.18 INVALID_TYPE = -24

Error Value:

INVALID_TYPE = -24

Error Description:

- This error code is returned when the “type” field following the requested command is illegal.

13.2.19 INVALID_PARAM = -25

Error Value:

INVALID_PARAM = -25

Error Description:

- This error code is returned when the “param” field following the requested command is illegal.

13.2.20 INVALID_PATH = -26

Error Value:

INVALID_PATH = -26

Error Description:

- This error code indicates that there is no such file or directory.

13.2.21 DIR_EXIST = -27

Error Value:

DIR_EXIST = -27

Error Description:

- This error code indicates that the directory specified by **AMBA_MKDIR** already exists so the execution of the command is unsuccessful.

13.2.22 PERMISSION_DENIED = -28

Error Value:

PERMISSION_DENIED = -28

Error Description:

This error code indicates that the search/write permission is denied for the specified path.

13.2.23 AUTHENTICATION_FAILED = -29

Error Value:

AUTHENTICATION_FAILED = -29

Error Description:

This error code indicates that the password does not match the authentication.

13.2.24 CARD_REMOVED = -30

Error Value:

CARD_REMOVED = -30

Error Description:

This error code indicates that the SD card has been removed so it is not available for dedicated operations such as take photo, start record, and so on.

Appendix 1 RTSP playback streaming

Ambarella A-Series wireless connectivity not only supports an over-WiFi RTSP stream for View Finder, please refer to Section 5.2.8, but also provides an over-WiFi RTSP file playback streaming function. The user can request a RTSP file streaming to playback by an RTSP URL, which names the files stored in our device or SD card.

Steps for an application to playback a file in the device through the RTSP protocol:
Steps 1 ~ 4 describes how to get all available file names.
Steps 5 describes what requests should be send from a RTSP client.

1. Connect to the WiFi command connection. (Please refer to Section 2.4 and Section 2.9).

   Send AMBA_START_SESSION command to start a control session. (Please refer to Section 4.2.1).
   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 257,"token" : 0}
   ```
   Remote Controller : Device:
   ```
   { "rval": 0, "msg_id": 257, "param": 1 }
   ```

2. Send **AMBA_CD** command to change the directory to the target folder. (Please refer to Section 7.2.3)

   Example:
   ```
   Remote Controller -> Device:
   {"msg_id" : 1283,"param" : "/tmp/fuse_d","token" : 1}
   Remote Controller : Device:
   {"rval": 0, "msg_id": 1283, "pwd": "/tmp\/fuse_d" }
   ```

3. Send **AMBA_LS** command to get the available file list of the current directory. (Please refer to Section 7.2.2)

   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 1282, "token" : 1}
   ```
   Remote Controller <- Device:
   ```
   {"rval":0,"msg_id":1282,"listing":[{"MISC/":"2013-08-05
   08:02:52"},{"DCIM/":"2013-07-30 11:24:26"},{"0. MP4":"2013-07-18
   10:41:48"},{"30fps.MP4":"2008-01-27 13:44:10"},{"1.MP4":"2008-01-
   29 13:54:19"}]}
   ```
   The item in the "listing" array is a name of a file or a directory, which has a trailing "/". In this example, MISC and DCIM are folders.

4. Now the user has the target file path, and a RTSP URL looks like: "rtsp://<Server's IP>/[File Full Path]".

   Example:
   ```
   rtsp://192.168.42.1/tmp/fuse_d/1.MP4 or
   rtsp://192.168.42.1/tmp/fuse_d/DCIM/100MEDIA/AMBA0001.MP4
   ```

5. A RTSP client has responsibility to start up a RTSP session with the server and send out a RTSP request format based on this RTSP URL. To start the file streaming, the RTSP client should send out OPTIONS, DESCRIBE, SETUP and PLAY methods sequentially. When the RTSP client want to stop the streaming, it should issue PAUSE and TEARDOWN methods and end the RTSP session. During the file streaming, the RTSP client should send out any command supported in OPTIONS to the server, such as **GET_PARAMETER**, to let the RTSP server know that the connection is still active.

   Following are examples for these RTSP requests and the reply from server.

Options:

Client request è

```
OPTIONS rtsp://192.168.42.1/tmp/fuse_d/1.MP4 RTSP/1.0
CSeq: 1
        User-Agent: APPPLAYER
```

Table A1-1. Client Request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 1
          Date:  Thu, Jan 01 1970 00:01:16 GMT
          Public:  OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PA-
          RAMETER.
```

Table A1-2. Server reply.

Describe:

Client request è

```
DESCRIBE rtsp://192.168.42.1/tmp/fuse_d/1.MP4 RTSP/1.0
CSeq: 2
        Accept: application/sdp
        User-Agent: APPPLAYER
```

Table A1-3. Client Request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 2
        Date: Thu, Jan 01 1970 00:01:17 GMT
        Content-Base: rtsp://192.168.42.1/tmp/fuse_d/1.MP4/
        Content-Type: application/sdp
        Content-Length: 716

v=0
        o=- 76830000 1 IN IP4 192.168.42.1
        s=Ambarella streaming
        i=Ambarella streaming
        t=0 0
        a=tool:Ambarella streaming 2012.03.12
        a=type:broadcast
        a=control:*
        a=range:npt=0-
        a=x-qt-text-nam:Ambarella streaming
        a=x-qt-text-inf:Ambarella streaming
        m=video 0 RTP/AVP 96
        c=IN IP4 0.0.0.0
        b=AS:10000
        a=rtpmap:96 H264/90000
        a=fmtp:96 packetization-mode=1;profile-level-id=4D401F;sprop-parameter-sets=J01AH5pkBsHuUvgPs
        gAAB9IAAdTB0IAC80AAF5oF3IxoQAF5oAALzQLvLhQ=,KO48gA==
        a=control:track1
        m=audio 0 RTP/AVP 97
        b=AS:1000
        a=rtpmap:97 MPEG4-GENERIC/48000/2
        a=fmtp:97 streamtype=5;profile-level-id=1;mode=AAC-hbr;sizelength=13;indexlength=3;indexdeltalengt
        h=3;config=1190
        a=control:track2
```

Table A1-4. Server reply.

SETUP:

In response to the DESCRIBE command from the server, the client receives the number of tracks. If the file includes video and audio streams, the user has to SETUP two media streams.

Client request è

```
SETUP rtsp://192.168.42.1/tmp/fuse_d/1.MP4/track1 RTSP/1.0
CSeq: 3
      Transport: RTP/AVP;unicast;client_port=63198-63199
      User-Agent: APPPLAYER
```

Table A1-5. Client Request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 3
     Date: Thu, Jan 01 1970 00:01:17 GMT
     Transport:
     RTP/AVP;unicast;destination=192.168.42.3;source=192.168.42.1;client_port=63198-631
     99;server_port=6970-6971
     Session: 82B26E0B
```

Table A1-6. Server reply.

Client request è

```
SETUP rtsp://192.168.42.1/tmp/fuse_d/1.MP4/track2 RTSP/1.0
CSeq: 4
      Transport: RTP/AVP;unicast;client_port=63200-63201
      Session: 82B26E0B
      User-Agent: APPPLAYER
```

Table A1-7. Client request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 4
     Date: Thu, Jan 01 1970 00:01:17 GMT
     Transport: RTP/AVP;unicast;destination=192.168.42.3;source=192.168.42.1;client_port=63200- 63201;serv-
er_port=6972-6973
     Session: 82B26E0B
```

Table A1-8. Server reply.

PLAY:

Client request è

```
PLAY rtsp://192.168.42.1/tmp/fuse_d/1.MP4/ RTSP/1.0
CSeq: 5
     Session: 82B26E0B
     Range: npt=0.000-
     User-Agent: APPPLAYER
```

Table A1-9. Client request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 5
     Date: Thu, Jan 01 1970 00:01:18 GMT
     Range: npt=0.000-
     Session: 82B26E0B
     RTP-Info:
     url=rtsp://192.168.42.1/tmp/fuse_d/1.MP4/track1;seq=49566;rtptime=7672614,url=rtsp://
     192.168.42.1/tmp/fuse_d/1.MP4/track2;seq=50031;rtptime=4092061
```

Table A1-10. Server reply.

PAUSE:

Client request è

```
PAUSE rtsp://192.168.42.1/tmp/fuse_d/1.MP4/ RTSP/1.0
CSeq: 6
     Session: 82B26E0B
     User-Agent: APPPLAYER
```

Table A1-11. Client request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 6
      Date: Thu, Jan 01 1970 00:01:22 GMT
      Session: 82B26E0B
```

Server reply.

TEARDOWN:

Client request è

```
TEARDOWN rtsp://192.168.42.1/tmp/fuse_d/1.MP4/ RTSP/1.0
CSeq: 7
      Session: 82B26E0B
     User-Agent: APPPLAYER
```

Table A1-12. Client request è.

Server reply:

```
RTSP/1.0 200 OK
CSeq: 7
      Date: Thu, Jan 01 1970 00:01:29 GMT
```

Table A1-13. Server reply.

Appendix 2 AMBA_GET_FILE

Remote device could use AMBA_GET_FILE to download the multimedia file and then play it or upload to social network. The file transfer port is 8787. Once the camera receives **AMBA_GET_FILE** command and the file is found, the camera begins to write data to the client who connect() to port 8787. Currently only one transfer at a time is allowed.

1. Connect to the WiFi command connection. (Please refer to Section 2.4 and Section 2.9).
2. Send **AMBA_START_SESSION** command to start a control session. (Please refer to Section 4.2.1).

   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 257,"token" : 0}
   ```
   Remote Controller <- Device:
   ```
   {"rval": 0, "msg_id": 257, "param": 1}
   ```

3. Send **AMBA_SET_CLNT_INFO** command to specify remote client information. (Please refer to Section x.x.x).

   Example: (For WIFI)
   Remote Controller -> Device:
   ```
   {"token": 1,"msg_id":261,"type": "TCP","param": "192.168.42.5"}
   ```
   Remote Controller <- Device:
   ```
   {"rval": 0,"msg_id": 261}
   ```

   Example: (For RFCOMM)
   Remote Controller -> Device:
   ```
   {"token": 1,"msg_id": 261,"type":"RFCOMM","param":"a0:b1:c2:d3:e4:f5"}
   ```
   Remote Controller <- Device:
   ```
   {"rval": 0,"msg_id": 261}
   ```

4. Send **AMBA_CD** command to change directory to the target folder. (Please refer to Section 7.2.3)

   Example:
   Remote Controller -> evice:
   ```
   {"msg_id" : 1283,"param" : "/tmp/fuse_d","token" : 1}
   ```
   Remote Controller <- Device:
   ```
   {"rval": 0, "msg_id": 1283, "pwd":"/tmp\/fuse_d"}
   ```
5. Send AMBA_LS command to get the available file list of the current directory (Please refer to Section 7.2.2)

   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 1282,"token" : 1}
   ```
   Remote Controller <- Device:
   ```
   {"rval":0,"msg_id":1282,"listing":[{"MISC/":"2013-08-05
   08:02:52"},{"DCIM/":"2013-07-30 11:24:26"},{"0.MP4":"2013-07-18
   10:41:48"},{"30fps.MP4":"2008-01-27 13:44:10"},{"1.MP4":"2008-01-29
   13:54:19"}]}
   ```
   The item in the "listing" array is a name of a file or a directory, which ends with a "/".
   In this example, MISC and DCIM are folders.

6. Send **AMBA_GET_FILE** command to download the file.

   Example:
   Remote Controller -> Device:
   {"fetch_size" : 0,"msg_id" : 1285,"offset" : 0,"param" : "AMBA0165.jpg","token" : 1}
   Remote Controller <- Device:
   {"rval":0,"msg_id":1285,"rem_size":1082297,"size":1082297}

7. Client then begins to retrieve the file from port 8787.

   When the file download is complete, the camera will send {"msg_id": 7, "type": "get_file_complete" ,"param":[{"bytes sent":1082297},{"md5sum":"f451477b2af009ca04a2733446edeffe"}]} to client

8. Client should verify the m5dsum with the downloaded file.
9. Client device then plays the file after file download done.

Appendix 3 HTTP Download

If the HTTP server has been enabled on the camera, the client device could download the multimedia file from the camera through HTTP GET command and then play it. Some browsers, such as Safari on iOS, support progressive download operation which could start playback while the file is downloading.

This is an example flow for playback with HTTP download:

1. Connect to the WiFi command connection. (Please refer to Section 2.4 and Section 2.9)

   Send **AMBA_START_SESSION** command to start a control session. (Please refer to Section 4.2.1).
   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 257,"token" : 0}
   ```
   Remote Controller <- Device:
   ```
   { "rval": 0, "msg_id": 257, "param": 1 }
   ```

2. Send AMBA_CD command to change directory to the target folder. (Please refer to Section 7.2.3)

   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 1283,"param" : "/tmp/fuse_d","token" : 1}
   ```
   Remote Controller <- Device:
   ```
   {"rval": 0, "msg_id": 1283, "pwd": "/tmp\/fuse_d"}
   ```

3. Send **AMBA_LS** command to get the available file list of the current directory (Please refer to Section 7.2.2)

   Example:
   Remote Controller -> Device:
   ```
   {"msg_id" : 1282,"token" : 1}
   ```
   Remote Controller <- Device:
   ```
   {"rval":0,"msg_id":1282,"listing":[{"MISC/":"2013-08-05
   08:02:52"},{"DCIM/":"2013-07-30 11:24:26"},{"0.MP4":"2013-07-18
   10:41:48"},{"30fps.MP4":"2008-01-27 13:44:10"},{"1.MP4":"2008-01-29
   13:54:19"}]}
   ```
   The item in the "listing" array is a name of a file or a directory, which ends with a "/". In this example, MISC and DCIM are folders.

4. Now that the user has the target file path, and the HTTP link for the target file looks like "http://<Server's IP>/[File Path]".

   Example:
   ```
   http://192.168.42.1/tmp/fuse_d/1.MP4
   ```
   or
   ```
   http://192.168.42.1/DCIM/100MEDIA/AMBA0001.MP4
   ```

5. The client device uses HTTP GET to download the file then play it or fill the link onto browser and then performs progressive download to play it during downloading.

Appendix 4 Examples of AMBA_GET_SETTING and AMBA_SET_SETTING

Description:

For different camera applications, such as Car-DV and Sport-DV, only dedicated setting types and options based on specifications are supported. In this appendix, Ambarella lists some examples for user reference. Please design suitable settings based on the requirements.

A4.1 Example: AMBA_GET_SETTING -> STATUS

- **Description**: This command is used to gather the current status of the camera.
- **Direction**: From the handheld to the Camera
- **Returns**: This command returns the camera status without altering the operational mode.

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1	Remote command number (values must be decimal)
type	"app_status"	Current status of the camera

*Table A4-1. Arguments for System Command API **AMBA_GET_SETTING > STATUS**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success < 0: Error as defined in Chapter 13
msg_id	1	The **rval** is the response to this remote command
type	"app_status"	Current status of the camera
param	CameraStatus	Status of the camera

*Table A4-2. Returns for System Command API **AMBA_GET_SETTING > STATUS**.*

Example:

- Arguments: **{"token": TokenNumber, "msg_id": 1, "type": "app_status"}**
- Returns for Camera Status = ViewFinder: {**"rval": 0, "msg_id": 1, "type": "app_status", "param": "vf "**}
- Status possible options: vf(view_finder), idle, record(recording_video), capture, photo_mode

A4.2 Example: AMBA_GET_SETTING -> CAMERA_CLOCK

- **Description**: This command is used to retrieve the camera's date and time.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	1	Remote command number (values must be decimal)
type	"camera_clock"	Name of the parameter to set

*Table A4-3. Arguments for System Command API **AMBA_GET_SETTING > CAMERA_CLOCK**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	1	The **rval** is the response to this remote command
type	"camera_clock"	Name of the parameter to set
param	<YYYY-MM-DD HH:MM:SS>	Time including Year (Y), Month (M), Day (D), Hours (H), Minutes (M) and Seconds (S). All fields must be accurately filled in. Any omission will void the command. For example, 2012-12-24 23:59:59 (24-hour format) represents one second before midnight on 24 December 2012.

*Table A4-4. Returns for System Command API **AMBA_GET_SETTING > CAMERA_CLOCK**.*

Example:

- Arguments to retrieve the camera's clock and date: **{"token": TokenNumber, "msg_id": 1, "type": "camera_clock"}**
- Successful return: **{"rval": 0, "msg_id": 1, "type": "camera_clock", "param": "2012-12-24 23:59:59"}**

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.3 Example: AMBA_GET_SETTING -> EXT_GPS

- **Description**: This command is used to control whether the camera will use GPS-related information from the handheld device or not. When this setting is enabled, the camera will use GPS information provided by the handheld device during recording. The handheld device is obligated to provide GPS-related information by using **AMBA_PUT_GPS_INFO** when **EXT_GPS** is enabled. Otherwise, the GPS information referenced by the camera may be incorrect.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"ext_gps"	Name of the parameter to set
param	ParameterValueToChange	Use external GPS information "on": Enable

*Table A4-5. Arguments for System Command API **AMBA_GET_SETTING > EXT_GPS**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"ext_gps"	Name of the parameter to set

*Table A4-6. Returns for System Command API **AMBA_GET_SETTING > EXT_GPS**.*

Example:

- Arguments to set the camera's clock and date: **{"token": TokenNumber, "msg_id": 2, "type": "ext_gps", "param": "on"}**
- Successful return: **{"rval": 0, "msg_id": 2, "type": "ext_gps"}**

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.1 "AMBA_GET_SETTING"
- Chapter 13 "System Errors"

A4.4 Example: AMBA_GET_SETTING -> MICROPHONE

- **Description**: This command is used to turn on/off microphone. Note that the phone-App should not display "mute icon" if **AMBA_SET_SETTING** -> **MICROPHONE** is absent in **AMBA_GET_ALL_CURRENT_SETTING**.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"microphone"	Name of the parameter to set
param	ParameterValueToChange	Enable/Disable MicroPhone "on": Enable

*Table A4-7. Arguments for System Command API **AMBA_GET_SETTING > MICROPHONE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"microphone"	Name of the parameter to set

*Table A4-8. Returns for System Command API **AMBA_GET_SETTING > MICROPHONE**.*

Example:

- Arguments to set the camera's clock and dat: **{"token": TokenNumber, "msg_id": 2, "type": "microphone", "param": "off"}**
- Successful return: {**"rval": 0, "msg_id": 2, "type": "microphone"**}

A4.5 Example: AMBA_GET_SETTING -> LANGUAGE

- **Description**: This command is used to select the language.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	“language”	Name of the parameter to set
param	ParameterValueToChange	Language Options: ”English”, ”Simplified_Chinese”, ”Traditional_Chinese” and ”Russian”

*Table A4-9. Arguments for System Command API **AMBA_GET_SETTING > LANGUAGE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	“language”	Name of the parameter to set

*Table A4-10. Returns for System Command API **AMBA_GET_SETTING > LANGUAGE**.*

Example:

- Arguments to set the camera’s clock and date: **{“token”: TokenNumber, “msg_id”: 2, “type”: “language”, “param”: “English”}**
- Successful return: {**“rval”: 0, “msg_id”: 2, “type”: “language”**}

A4.6 Example: AMBA_GET_SETTING -> DEFAULT_SETTING

- **Description**: This command is used to reset the camera setting.
- Note that phone-App will not display the “reset icon” if this **AMBA_SET_SETTING** -> **DEFAULT_SETTING** is absent in **AMBA_GET_ALL_CURRENT_SETTING**.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	“default_setting”	Name of the parameter to set
param	ParameterValueToChange	Options: ”on”: Reset ”off”: Do nothing

*Table A4-11. Arguments for System Command API **AMBA_GET_SETTING > DEFAULT_SETTING**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	“default_setting”	Name of the parameter to set

*Table A4-12. Returns for System Command API **AMBA_GET_SETTING > DEFAULT_SETTING**.*

Example:

- Arguments to set the microphone mute: **{“token”: TokenNumber, “msg_id”: 2, “type”: “default_setting”, “param”: “on”}**
- Successful return: **{“rval”: 0, “msg_id”: 2, “type”: “default_setting”}**

A4.7 Example: AMBA_GET_SETTING -> VIDEO_RESOLUTION

- **Description**: This command is used to configure the video resolution (width) x (height) (frame rate) (aspect ratio) as described here.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"video_resolution"	Name of the parameter to set
param	ParameterValueToChange	Video Resolution (width) x (height) (frame rate) (aspect ratio) Example video resolutions for A5s are as follows: "1920x1080 30P 16:9" "1280x720 60P 16:9" "1280x720 30P 16:9"

*Table A4-13. Arguments for System Command API **AMBA_GET_SETTING** > **VIDEO_RESOLUTION**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"video_resolution"	Name of the parameter to set

*Table A4-14. Returns for System Command API **AMBA_GET_SETTING** > **VIDEO_RESOLUTION**.*

Example:

- Arguments for video resolution 1280x720p60: **{"token": TokenNumber, "msg_id": 2, "type": "video_resolution", "param": "1280x720 60P 16:9"}**
- Successful return: {**"rval": 0, "msg_id": 2, "type": "video_resolution"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.8 Example: AMBA_GET_SETTING -> VIDEO_QUALITY

- **Description**: This command is used to define the video quality.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"video_quality"	Name of the parameter to set
param	ParameterValueToChange	Video quality definition: Example video qualities for A5s: "sfine": Super fine "fine": Fine

*Table A4-15. Arguments for System Command API **AMBA_GET_SETTING > VIDEO_QUALITY**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"video_quality"	Name of the the parameter to set

*Table A4-16. Returns for System Command API **AMBA_GET_SETTING > VIDEO_QUALITY**.*

Example:

- Arguments for video quality fine: **{"token": TokenNumber, "msg_id": 2, "type": "video_ quality", "param": "fine"}**
- Successful return: {**"rval": 0, "msg_id": 2, "type": "video_quality"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.9 Example: AMBA_GET_SETTING -> SAVE_LOW_RESOLUTION_CLIP

- **Description**: It is used to enable/disable a second standard-definition (SD) video. When enabled, the lower-version (compared to high-resolution) video will be recorded to the camera's storage. This command will take effect on the next start record.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"save_low_resolution_clip"	Name of the parameter to set
param	ParameterValueToChange	Lower resolution video recording enable / disable: "on": Enable

*Table A4-17. Arguments for System Command API **AMBA_GET_SETTING > SAVE_LOW_RESOLUTION_CLIP**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"save_low_resolution_clip"	The lower-resolution record

*Table A4-18. Returns for System Command API **AMBA_GET_SETTING > SAVE_LOW_RESOLUTION_CLIP**.*

Example:

- Arguments for video quality fine: **{"token": TokenNumber, "msg_id": 2, "type": "save_low_resolution_clip", "param": "on"}**
- Successful return: **{"rval": 0, "msg_id": 2, "type": "save_low_resolution_clip"}**

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.10 Example: AMBA_GET_SETTING -> STREAM_OUT_TYPE

- **Description**: The command is used to set the Remote View Finder streaming protocol (e.g., RTSP, MJPG or HLS). The command takes effect on the next enabling of the remote View Finder. The transmission of View Finder is disabled if this API is set as "none".

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"stream_out_type"	Name of the parameter to set
param	ParameterValueToChange	Remote View Finder Types: Options are: "none": Disable the transmission of the View Finder

*Table A4-19. Arguments for System Command API **AMBA_GET_SETTING > STREAM_OUT_TYPE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"stream_out_type"	Name of the parameter to set

*Table A4-20. Returns for System Command API **AMBA_GET_SETTING > STREAM_OUT_TYPE**.*

Example 1:

- Arguments for video quality fine: **{"token": TokenNumber, "msg_id": 2, "type": "stream_out_type", "param": "rtsp"}**
- Successful return: {**"rval": 0, "msg_id": 2, "type": "stream_out_type"**}

Example 2:

- Enable view finder Streaming Protocol to MJPG and record both HD and SD videos

- To set view finder streaming protocol to "MJPG" requires a set of command operations from the handheld device: Enable save_low_resolution_clip and set stream_out_type to MJPG.

- Arguments to Enable Standard Definition Video: {**"token": TokenNumber, "msg_id": 2, "type": "save_low_resolution_clip", "param": "on"**}

- Arguments to set view finder to Type MJPG: {**"token": TokenNumb er, "msg_id": 2, "type":**

“stream_out_type”, “param”: “mjp g”}

- On successful command operations, the MJPG stream can be retrieved over WiFi via HTTP GET from: http://<Camera IP>/mjpeg/amba.jpg.

Example 3:

- Set the remote view finder Streaming Protocol to RTSP without SD recording
- To set the streaming protocol to RTSP requires a set of command operations from the handheld device: Disable save_low_resolution_clip and set stream_type to RTSP.
- Arguments to Disable Standard Definition Video: {**“token”: TokenNumber, “msg_id ”: 2, “type”: “save_low_resolution_clip”, “param”: “off”**}
- Arguments for setting Viewfinder to Type RTSP: {**“token”: TokenNumb er, “msg_id ”: 2, “type”: “stream_out_type”, “param”: “rtsp”**}
- On successful command operations, an RTSP stream can be retrieved over WiFi from: rtsp://<Camera IP>/live.

See Also:

- Section 2.6 “Design: Configuration of Settings and Capabilities”
- Section 4.2.1 “AMBA_START_SESSION”
- Section 3.2.2 “AMBA_SET_SETTING”
- Chapter 13 “System Errors”

A4.11 Example: AMBA_GET_SETTING -> MULTI_CHANNEL

- **Description**: This command is used to enable / disable specific channels.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"multi_channel"	Name of the parameter to set
channel_num	<channel number>	Specify which channel to turn on / off.
param	<Boolean>	Indicate whether a specific channel is on/off: "on": Enable

*Table A4-21. Arguments for System Command API **AMBA_GET_SETTING > MULTI_CHANNEL**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"multi_channel"	Name of the parameter to set
channel_num	<channel number>	Indicate which channel to turn on / off

*Table A4-22. Returns for System Command API **AMBA_GET_SETTING > MULTI_CHANNEL**.*

Example:

- Arguments: {**"token": TokenNumber, "msg_id": 2, "type":"multi_channel", "channel_num": 3, "param": "on"**} Turn on channel 3
- Successful return: {**"rval": 0, "msg_id": 2, "type": "multi_channel", "channel_num":3**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.12 Example: AMBA_GET_SETTING -> VIDEO_LOG

- **Description**: This command is used to enable / disable the video log feature.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"video_log"	Name of the parameter to set
param	ParameterValueToChange	Enable/Disable Video Log: "on": Enable

*Table A4-23. Arguments for System Command API **AMBA_GET_SETTING** > **VIDEO_LOG**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"video_log"	Name of the parameter to set

*Table A4-24. Returns for System Command API **AMBA_GET_SETTING** > **VIDEO_LOG**.*

Example:

- Arguments: {**"token": TokenNumber, "msg_id": 2, "type":"video_log", "param": "on"**}
- Successful return: {**"rval": 0, "msg_id": 2, "type": "video_log"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.13 Example: AMBA_GET_SETTING -> VIDEO_LOG_DURATION

- **Description**: This command is used to set the duration of each video clip in a video log.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"video_log_duration"	Name of the parameter to set
param	<Duration In Seconds>	The duration length setting should be consistent with the list of available settings obtained by using **AMBA_GET_SINGLE_SETTING_OPTIONS** with the **video_log_duration** argument.

*Table A4-25. Arguments for System Command API **AMBA_GET_SETTING > VIDEO_LOG _DURATION**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"video_log_duration"	Name of the parameter to set

*Table A4-26. Returns for System Command API **AMBA_GET_SETTING > VIDEO_LOG _DURATION**.*

Example:

- Arguments: {**"token": TokenNumber, "msg_id": 2, "type":"video_log_duration", "param": "10"**}
- Successful return: {**"rval": 0, "msg_id": 2, "type": "video_log _duration"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.14 Example: AMBA_GET_SETTING -> VIDEO_LOG_INTERVAL

- **Description**: This command is used to set the interval length between video clips in a video log.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"video_log_duration"	Name of the parameter to set
param	<Interval In Seconds>	The duration length setting should be consistent with the list of available settings obtained by using **AMBA_GET_SINGLE_SETTING_OPTIONS** with the **video_log_Interval** argument.

*Table A4-27. Arguments for System Command API **AMBA_GET_SETTING > VIDEO_LOG _INTERVAL**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"video_log_interval"	Name of the parameter to set

*Table A4-28. Returns for System Command API **AMBA_GET_SETTING > VIDEO_LOG _INTERVAL**.*

Example:

- Arguments: {**"token": TokenNumber, "msg_id": 2, "type":"video_log_interval", "param": "60"**}
- Successful return: {**"rval": 0, "msg_id": 2, "type": "video_log _interval"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.15 Example: AMBA_GET_SETTING -> TIMELAPSE_VIDEO

- **Description**: This command is used to turn on/off Time Lapse video. **AMBA_RECORD_START** will trigger time-lapse video recording when “timelapse_video!=off”.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	“timelapse_video”	Name of the parameter to set
param	ParameterValueToChange	Interval length in seconds (sec). It should be in a string form and one of these values (depends on the result from **AMBA_GET_SINGLE_SETTING_OPTIONS**): “off”, “2”, “5”, “10”, “30”.

*Table A4-29. Arguments for System Command API **AMBA_GET_SETTING > TIMELAPSE_VIDEO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	“timelapse_video”	Name of the parameter to set

*Table A4-30. Returns for System Command API **AMBA_GET_SETTING > TIMELAPSE_VIDEO**.*

Example:

- Arguments: {**“token”: TokenNumber, “msg_id”: 2, “type”: “timelapse_video”, “param” : “2”**}

- Successful return: {**“rval”: 0, “msg_id”: 2, “type”: “timelapse_video”**}

See Also:

- Section 3.2.1 “AMBA_GET_SETTING”
- Section 4.2.1 “AMBA_START_SESSION”
- Section 3.2.2 “AMBA_SET_SETTING”
- Chapter 13 “System Errors”

A4.16 Example: AMBA_GET_SETTING -> CAPTURE_MODE

- **Description**: This command is used to specify the photo capture mode.
- Different photo capture modes may have different image size options. This leads to dependencies between modes and image sizes. Please retrieve options for photo size after changing the **capture_mode**.
- Note that some of the camera operations can only be enabled at certain specific **capture_mode**. For example, the user must set **capture_mode** as "precise cont." so that time-lapse photo setting can be adjusted (otherwise, this setting will be read-only). Furthermore, when a time-lapse photo is enabled, the **capture_mode** is locked at "precise cont.". Please disable the time-lapse photo feature, if the user intends to change the **capture_mode**.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"capture_mode"	Name of the parameter to set
param	ParameterValueToChange	Capture mode: Examples: "precise quality": Precise single shot capture

*Table A4-31. Arguments for System Command API **AMBA_GET_SETTING > CAPTURE_MODE**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"capture_mode"	Name of the parameter to set

*Table A4-32. Returns for System Command API **AMBA_GET_SETTING > CAPTURE_MODE**.*

Example:

- Arguments for precise quality capture mode: {**"token": TokenNumber, "msg_id": 2, "type":"capture_mode", "param": "precise quality"**}

- Successful return: {**"rval": 0, "msg_id": 2, "type": "capture_mode"**}

See Also:

- Section 2.6 “Design: Configuration of Settings and Capabilities”
- Section 4.2.1 “AMBA_START_SESSION”
- Section 3.2.2 “AMBA_SET_SETTING”
- Chapter 13 “System Errors”

A4.17 Example: AMBA_GET_SETTING -> TIMELAPSE_PHOTO

- **Description**: This command is used to turn on/off the Time Lapse Photo function. **AMBA_TAKE_PHOTO** will trigger time-lapse photo when "timelapse_photo=on".

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"timelapse_photo"	Name of the parameter to set
param	ParameterValueToChange	Interval length in seconds (sec). It should be in a string form and one of these values (depends on the result from **AMBA_GET_SINGLE_SETTING_OPTIONS**): "off", "2", "5", "10", "30".

*Table A4-33. Arguments for System Command API **AMBA_GET_SETTING > TIMELAPSE_PHOTO**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"timelapse_photo"	Name of the parameter to set

*Table A4-34. Returns for System Command API **AMBA_GET_SETTING > TIMELAPSE_PHOTO**.*

Example:

- Arguments for precise quality capture mode: {**"token": TokenNumber, "msg_id": 2, "type":"timelapse_photo", "param": "2"**}

- Successful return: {**"rval": 0, "msg_id": 2, "type": "timelapse_photo"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

A4.18 Example: AMBA_GET_SETTING -> PHOTO_LOG

- **Description**: This command is used to measure interval length to be used during the Time Lapse Photo.

- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	“photo _log”	Name of the parameter to set
param	ParameterValueToChange	Enable/Disable Photo Log: “on”: Enable

*Table A4-35. Arguments for System Command API **AMBA_GET_SETTING > PHOTO_LOG**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	“photo_log”	Name of the parameter to set

*Table A4-36. Returns for System Command API **AMBA_GET_SETTING > PHOTO_LOG**.*

Example:

- Arguments to turn on photo_log: {**“token”: TokenNumber, “msg_id”: 2, “type”:“photo_log”, “param”: “on”**}

- Successful return: {**“rval”: 0, “msg_id”: 2, “type”: “photo_log”**}

See Also:

- Section 2.6 “Design: Configuration of Settings and Capabilities”
- Section 4.2.1 “AMBA_START_SESSION”
- Section 3.2.2 “AMBA_SET_SETTING”
- Chapter 13 “System Errors”

A4.19 Example: AMBA_GET_SETTING -> PHOTO_LOG_INTERVAL

- **Description**: It is used to specify the interval length of the **photo_log**.
- **Direction**: From the handheld to the Camera

Arguments:

Key	Value	Description
token	TokenNumber	Session ID returned by **AMBA_START_SESSION** (Section 4.2.1)
msg_id	2	Remote command number (values must be decimal)
type	"photo _log_interval"	Name of the parameter to set
param	ParameterValueToChange	Interval length in seconds (sec). It should be in a string form and one of these values: "10", "15", "30", "60", "90", "120", "300", "600", "1200", "1800", "3600".

*Table A4-37. Arguments for System Command API **AMBA_GET_SETTING** > **PHOTO_LOG_INTERVAL**.*

Returns:

Key	Value	Description
rval	≤ 0	0: Success <0: Error as defined in Chapter 13
msg_id	2	The **rval** is the response to this remote command
type	"photo_log_interval"	Name of the parameter to set

*Table A4-38. Returns for System Command API **AMBA_GET_SETTING** > **PHOTO_LOG_INTERVAL**.*

Example:

- Arguments to turn on photo_log: {**"token": TokenNumber, "msg_id": 2, "type":"photo_log_interval", "param": "10"**}
- Successful return: {**"rval": 0, "msg_id": 2, "type": "photo_log_interval"**}

See Also:

- Section 2.6 "Design: Configuration of Settings and Capabilities"
- Section 4.2.1 "AMBA_START_SESSION"
- Section 3.2.2 "AMBA_SET_SETTING"
- Chapter 13 "System Errors"

Appendix 5 Naming Rule of Files in the Default Camera after API 4.1.0

The specially naming rule is defined in the Camera after API 4.1.0. However, the following JSON structure is based on the currently naming rule in the Camera. If the naming rule is modified, please consider whether the application in the handheld can support to analyze the new design or not. The JSON structure below is just used to describe the definition, please use the correct JSON format such as the Example and combine it into the JSON string of **AMBA_GET_DEVICEINFO**.

The generally naming rule in the Camera after API 4.1.0

Definition:

There are some files which have different types and their naming rules are as below:

Section 1 + Section 2 + Section 3 + …

The JSON structure of naming rule can be defined as:

```
{“naming_rule”:
   [
      {
         “Type_Name”: “ File_Type_1 “
        “JSON Key 1” : [Begin_Index, String_Length],   /* Based on the Section 1*/
        “JSON Key 2” : [Begin_Index, String_Length], ],   /* Based on the Section 2*/
         “JSON Key 3” : [Begin_Index, String_Length], ],   /* Based on the s Section 3*/

  /* Please describe the naming rule here with key and JSON array which is used to indicate the beginning index and the string length.  The Camera can indicate any JSON_Key, Begin_Index and String_Length based on its design but please consider the handheld may need to use this key to do operation or analysis. */
        },

 {
          “Type_Name”: “ File_Type_2 “
       “JSON Key 1” : [Begin_Index, String_Length],   /* Based on the Section 1*/
        “JSON Key 2” : [Begin_Index, String_Length], ],   /* Based on the Section 2*/
         “JSON Key 3” : [Begin_Index, String_Length], ],   /* Based on the Section 3*/
            .
            .
            .

        /* Please describe the naming rule here with key and JSON array which is used to indicate the begin index and string length.  The camera can indicate any JSON_Key, Begin_Index and String_Length based on its design but please consider the handheld may need to use this key do operation or analysis. */

       }

     /* The camera can extend the content based on its design if it still has other file which has to describe the naming rule. Please define the type by yourself and consider the handheld can support it.*/

   ]
}
```

Example: The current implementation in the Camera for media files

Limitation:

In addition to the above description of naming rule, the user who uses the handheld may want to play any streaming with any sensor so that the handheld should get information of sensor and streaming. In other words, the camera has to provide the information in the JSON string for that to handheld. Media files in the current camera have two types and their file names are all based on the above naming rule.

The example of video name: 121212AB.mp4

121212 is the main name. A is used to represent a senor and B is used to represent a streaming.

The example of photo name: 12121201.jpg

121212 is the main name. 01 is used to represent the offset of picture file.

On the basis of the above rule and limitation, the default JSON string of naming rule is as below:

{“naming_rule”:[{“type”: “video”, “main_section”: [0,6], “sensor_section”:[6,1], “stream_section”:[7,1]},{“type”: “photo”,”main_section”: [0,6],”offset_section”:[6,2]}]}

NOTE: Even though the camera provides the information of sensor and streaming in the JSON string, the handheld may ignore them if its design does not need them.

Appendix 6 Extension for User-specific Command API

Please define the customized message id after “0x10000000”.

For example, the customer wants to define a new message “CUSTOMIZED_MSG_1” and its message ID can be defined as “0x10000001”.

Appendix 7 API List

API Group	API	API version
System Commands	**AMBA_GET_SETTING**	2.0.0
	AMBA_SET_SETTING	2.0.0
	AMBA_GET_ALL_CURRENT_SET-TINGS	2.0.0
	AMBA_GET_SINGLE_SETTING_OP-TIONS	2.0.0
	AMBA_FORMAT	2.0.0
	AMBA_GET_SPACE	2.0.0
	AMBA_GET_NUM_FILES	2.0.0
	AMBA_NOTIFICATION	2.0.0
	AMBA_BURNIN_FW	2.0.0
	AMBA_NOTIFICATION -> FW_UP-GRADE_COMPLETE	2.0.0
	AMBA_PUT_GPS_INFO	2.0.0
	AMBA_GET_DEVICEINFO	2.0.0
	AMBA_POWER_MANAGE	2.0.0
	AMBA_GET_BATTERY_LEVEL	2.0.0
	AMBA_SET_SETTING - > CAMERA_ CLOCK	2.0.0
	AMBA_GET_SETTING -> CAMERA_ CLOCK	2.0.0
	AMBA_SET_SETTING -> EXT_GPS	2.0.0
	AMBA_SET_SETTING -> MICRO-PHONE	4.0.0
	AMBA_SET_SETTING -> LANGUAGE	4.0.0
	AMBA_SET_SETTING -> DEFAULT_ SETTING	2.0.0
	AMBA_ZOOM	2.0.0
	AMBA_ZOOM_INFO	2.0.0
	AMBA_CHANGE_BITRATE	2.0.0
	AMBA_SET_PASSWD	4.0.0
Session Controls	**AMBA _START_SESSION**	2.0.0
	AMBA_STOP_SESSION	2.0.0
	AMBA_RESETVF	2.0.0
	AMBA_STOP_VF	2.0.0
	AMBA _GET_SETTING> STATUS	2.0.0
	AMBA_SET_CLNT_INFO	4.0.0
Video Commands	**AMBA _RECORD _START**	2.0.0
	AMBA-RECORD-STOP	2.0.0
	AMBA _GET_RECORD_TIME	2.0.0

	AMBA _FORCE_SPLIT	2.0.0
	AMBA_SET_SETTING> VIDEO _RESOLUTION	2.0.0
	AMBA_SET_SETTING> VIDEO_ QUALITY	2.0.0
	AMBA_SET_SETTING > SAVE _LOW_ RESOLUTION_CLIP	3.0.0
	AMBA_SET_SETTING > STREAM _OUT_TYPE	3.0.0
	AMBA_SET_SETTING > MULTI _CHANNEL	2.0.0
	AMBA_SET_SETTING > VIDEO _LOG	2.0.0
	AMBA_SET_SETTING > VIDEO _LOG_DURATION	2.0.0
	AMBA_SET_SETTING > VIDEO _LOG_INTERVAL	2.0.0
	AMBA_SET_SETTING > TIMELAPSE_ VIDEO	2.0.0
Photo Commands	**AMBA_TAKE_PHOTO**	2.0.0
	AMBA_ NOTIFICATION> CONTINUE_ BURST_COMPLETE	2.0.0
	AMBA_CONTINUE_CAPTURE _STOP	2.0.0
	AMBA_SET_SETTING > CAPTURE _MODE	2.0.0
	AMBA-SET-SETTING > TIMELAPSE_ PHOTO	2.0.0
	AMBA _SET_SETTING > PHOTO _LOG	2.0.0
	AMBA_SET_SETTING > PHOTO _LOG_INTERVAL	2.0.0
	AMBA_FOCUS	4.0.0
File System Commands	**AMBA-DEL-FILE**	2.0.0
	AMBA_LS	2.0.0
	AMBA_CD	2.0.0
	AMBA_PWD	2.0.0
	AMBA_GET_FILE	2.0.0
	AMBA_CANCEL_FILE_XFER	2.0.0
	AMBA_NOTIFICATION > GET_FILE _COMPLETE	2.0.0
	AMBA_PUT_FILE	2.0.0
	AMBA_NOTIFICATION > PUT_FILE _COMPLETE	2.0.0
	AMBA_MKDIR	4.0.0
WiFi Commands	**AMBA_WIFI_RESTART**	3.0.0
	AMBA_SET_WIFI_SETTING	3.0.0
	AMBA_GET_WIFI_SETIING	3.0.0
	AMBA_WIFI_STOP	3.0.0
	AMBA_WIFI_START	3.0.0
	AMBA_GET_WIFI_STATUS	3.0.0

Media Commands	**AMBA_GET_THUMB**	2.0.0
	AMBA_GET_MEDIAINFO	2.0.0
	AMBA_SET_MEDIA_ATTRIBUTE	2.0.0
Notifications	**AMBA_NOTIFICATION > DISCONNECT_HDMI**	2.0.0
	AMBA_NOTIFICATION > DISCONNECT_SHUTDOWN	2.0.0
	AMBA_NOTIFICATION > STARTING _VIDEO_RECORD	2.0.0
	AMBA_NOTIFICATION > VIDEO _RECORD_COMPLETE	2.0.0
	AMBA_NOTIFICATION > PHOTO _TAKEN	2.0.0
	AMBA_NOTIFICATION > CONTINUE _CAPTURE_START	2.0.0
	AMBA_NOTIFICATION > CONTINUE _CAPTURE_STOP	2.0.0
	AMBA_NOTIFICATION > CONTINUE _BURST_START	2.0.0
	AMBA-NOTIFICATION > CONTINUE BURST-COMPLETE	2.0.0
	AMBA_NOTIFICATION > LOW_BATTERY_WARNING	2.0.0
	AMBA_NOTIFICATION > LOW_STORAGE_WARNING	2.0.0
	AMBA_NOTIFICATION > TIMELAPSE _VIDEO_STATUS	2.0.0
	AMBA_NOTIFICATION > TIMELAPSE _PHOTO_STATUS	2.0.0
	AMBA_NOTIFICATION > CAMERA_ CONNECT_TO_PC	2.0.0
	AMBA_NOTIFICATION > LOG_UPDATED	2.0.0
	AMBA_NOTIFICATION > POWER_ MODE_CHANGE	2.0.0
	AMBA_NOTIFICATION > VF_START	2.0.0
	AMBA_NOTIFICATION > VF_STOP	2.0.0
	AMBA_NOTIFICATION > AUTO_FILE _ DELETE	2.0.0
	AMBA_NOTIFICATION > STORAGE _RUNOUT	2.0.0
	AMBA_NOTIFICATION > STORAGE _IO_ERROR	2.0.0
	AMBA_NOTIFICATION > LOW _ SPEED_CARD	2.0.0
	AMBA-NOTIFICATION > MUXER-INDEX_LIMIT	2.0.0
	AMBA_NOTIFICATION > MUXER _FILE_SIZE_LIMIT	2.0.0

	AMBA-NOTIFICATION > CARD-REMOVED	2.0.0
	AMBA_NOTIFICATION > CANNOT _ISSUE_PIV	2.0.0
	AMBA_NOTIFICATION > START_ EVENT_RECORD	4.0.0
	AMBA_NOTIFICATION -> EVENT_ RECORD_COMPLETE	4.0.0
	AMBA_NOTIFICATION -> FORCE_ DISCONNECT	4.0.0
	AMBA_NOTIFICATION -> MENU_ON	4.0.0
	AMBA_NOTIFICATION -> MENU_ OFF	4.0.0
	AMBA_NOTIFICATION -> THUMB_ MOTION_ON	4.0.0
	AMBA_NOTIFICATION -> THUMB_ MOTION_OFF	4.0.0
Query Commands	**AMBA_QUERY_SESSION_HOLDER**	2.0.0

Table A7-1. Different APIs in different SDK version.

Appendix 8 Additional Resources

Please contact an Ambarella representative for other documents of potential interest.

Appendix 9 Important Notice

All Ambarella design specifications, datasheets, drawings, files, and other documents (together and separately, "materials") are provided on an "*as is*" basis, and Ambarella makes no warranties, expressed, implied, statutory, or otherwise with respect to the materials, and expressly disclaims all implied warranties of noninfringement, merchantability, and fitness for a particular purpose. The information contained herein is believed to be accurate and reliable. However, Ambarella assumes no responsibility for the consequences of use of such information.

Ambarella Incorporated reserves the right to correct, modify, enhance, improve, and otherwise change its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

All products are sold subject to Ambarella's terms and conditions of sale supplied at the time of order acknowledgment. Ambarella warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent Ambarella deems necessary to support this warranty.

Ambarella assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using Ambarella components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

Ambarella does not warrant or represent that any license, either expressed or implied, is granted under any Ambarella patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Ambarella products or services are used. Information published by Ambarella regarding third-party products or services does not constitute a license from Ambarella to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Ambarella under the patents or other intellectual property of Ambarella.

Reproduction of information from Ambarella documents is not permissible without prior approval from Ambarella.

Ambarella products are not authorized for use in safety-critical applications (such as life support) where a failure of the product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Customers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of Ambarella products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Ambarella. Further, Customers must fully indemnify Ambarella and its representatives against any damages arising out of the use of Ambarella products in such safety-critical applications.

Ambarella products are neither designed nor intended for use in military/aerospace applications or environments. Customers acknowledge and agree that any such use of Ambarella products is solely at the Customer's risk, and they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Appendix 10 Revision History

NOTE: Page numbers for previous drafts may differ from page numbers in the current version.

Version	Date	Comments
1.0	8 Aug 2012	Formatting of original draft
1.1	28 Aug 2012	Update message ID (Section 3.2.3, Section 3.2.4)
1.2	15 Nov 2012	Expand to additional remote connections (e.g., Bluetooth, WiFi)
1.3	19 Nov 2012	Update description for error message CAMERA_OVERRIDE_FAIL (Ch 8)
1.4	27 Nov 2012	Add new AMBA_STOP_VF API (Ch 3) Update rval sign (Ch 2-6)
1.5		Update AMBA_GET_SETTING API (Ch 2)
1.6	5 Mar 2013	General update to command structure Add Notifications and Keywords chapters
1.7	7 Mar 2013	Update notifications chapter
2.0	17 May2013	Remove configuration file and use configuration database instead (Ch 2) Change message ID numbering in (all chapters)
2.1	22 May2013	Add put file and media commands
2.2	7 June 2013	Add section on design configuration for packet and packet headers (Ch 2) Update AMBA_GET_ALL_CURRENT_SETTINGS API (Ch 3) Update AMBA_GET_FILE API (Ch 7) Update AMBA_GET_THUMB (Ch 10)
2.3	12 June 2013	Remove Overview: SQLite (Ch 1) Update to Design: Configuration (Ch 2) Update to AMBA_GET_THUMB (Ch 10)
2.4	18 June 2013	Add command AMBA_GET_SETTING > SINGLE_SETTING_OPTIONS (Ch 3) Update AMBA_SET_SETTING > STD_DEF_VIDEO (Ch 5) Update AMBA_SET_SETTING > STREAM_TYPE (View Finder Protocol) (Ch 5) Update AMBA_SET_SETTING > TRANS_WHILE_RECORD (Ch 5)
2.5	20 June 2013	Change AMBA_GET_SETTING > SINGLE_SETTING_OPTIONS to AMBA_GET_SINGLE_SETTING_OPTIONS (Ch 3) Change AMBA_GET_RECORD_TIME return-type from string to integer Change AMBA_START SESSION return value to integer Change TRANS_WHILE_RECORD to STREAM_WHILE_RECORD
2.6	27 June 2013	Update AMBA_GET_SINGLE_SETTING_OPTIONS (Ch 3) Update AMBA_LS (Ch 7) Update AMBA_PUT_FILE (Ch 7) Update AMBA_NOTIFICATION >PUT_FILE_COMPLETE (Ch 7)
2.7	11 Jul 2013	Add AMBA_CANCEL_GET_FILE (Ch 7) Update AMBA_NOTIFICATION>GET_FILE_COMPLETE (Ch 7) Add AMBA_NOTIFICATION>CAMERA_CONNECT_TO_PC (Ch 9)
2.7.1	30 December 2013	Update in overview, Design, System, Session, Video and Photo commands. A new section Query added in between Media and System Errors. Refine all descriptions/formatting.
2.8.0	21 March 2014	Update in System, Session Control, Video, Notification, Error, Keywords and Photo. Add appendices.

Version	Date	Comments
2.8.1	5 June 2014	Update in **AMBA_CHANGE_BITRATE**, **AMBA_DIGITAL_ZOOM**, **AMBA_GET_BATTERY_LEVEL**, AMBA_SETTING -> capture_mode and **AMBA_MEDIA_INFO**.
3.0.0	11 August 2014	Update in Sections 3.2.12, 6.2.4, 7.2.6, 7.2.7, 7.2.8, 8.2.2, 8.2.3, 9.2.2, 10.2.16, and 10.2.23 Add Sections 3.2.18, 3.2.19, 3.2.20 (MICROPHONE, LANGUAGE, DEFAULT_SETTING APIs). Add new APIs to Chapter 8 (AMBA_SET_WIFI_SETTING, AMBA_GET_WIFI_SETTING, AMBA_WIFI_STOP, AMBA_WIFI_START & AMBA_GET_WIFI_STATUS).
4.0.0	4 March 2015	Update in Sections 3.2.9, 3.2.12, 3.2.20, 3.2.21, 4.2.1, 6.2.1, 6.2.3, 6.2.7, 7.2.5, 7.2.6, 7.2.8, 9.2.1 and Appendix 2. Added Sections 3.2.13 **AMBA_SET_PASSWD**, 4.2.2 **AMBA_SET_CLNT_INFO**, 6.2.8 **AMBA_FOCUS**, Appendix 4 API List, 7.2.10 **AMBA_MKDIR**, 10.2.27 **START_EVENT_RECORD**, 10.2.28 **EVENT_RECORD_COMPLETE**, 10.2.29 **FORCE_DISCONNECT**, 10.2.30 **MENU_ON**, 10.2.31 **MENU_OFF**, 10.2.32 **THUMB_MOTION_ON**, 13.2.21, 13.2.22 and 13.2.23.
4.1.0	10 June 2015	Updated in Sections 3.2.9, 3.2.12, 3.2.13, 7.2.6, 7.2.9, 9.2.1, 9.2.2 and 12.2. Added Appendix 4 Examples of AMBA_GET_SETTING and AMBA_SET_SETTING. Deleted Sections 3.2.16 AMBA_SET_SETTING > CAMERA CLOCK, 3.2.17 AMBA_GET_SETTING > CAMERA CLOCK, 3.2.18 AMBA_SET_SETTING > EXT_GPS, 3.2.19 AMBA_SET_SETTING > MICROPHONE, 3.2.20 AMBA_SET_SETTING > LANGUAGE, 3.2.21 AMBA_SET_SETTING > DEFAULT_SETTING, 4.2.6 AMBA_GET_SETTING > STATUS, 5.2.5 AMBA_SET_SETTING > VIDEO_RESOLUTION, 5.2.6 AMBA_SET_SETTING > VIDEO_QUALITY, 5.2.7 AMBA_SET_SETTING > VIDEO_SAVE_LOW_RESOLUTION_CLIP, 5.2.8 AMBA_SET_SETTING > STREAM_OUT_TYPE , 5.2.9 AMBA_SET_SETTING > MULTI_CHANNEL, 5.2.10 AMBA_SET_SETTING > VIDEO_LOG, 5.2.11 AMBA_SET_SETTING > VIDEO_LOG_DURATION, 5.2.12 AMBA_SET_SETTING > VIDEO_LOG_INTERVAL, 5.2.13 AMBA_SET_SETTING > TIMELAPSE_VIDEO, 6.2.4 AMBA_SET_SETTING > CAPTURE_MODE, 6.2.5 AMBA_SET_SETTING > TIMELAPSE_PHOTO and 6.2.6 AMBA_SET_SETTING > PHOTO_LOG, 6.2.7.

Table A10-1. Revision History.