Table of Contents

Log Management API

API that controls the device log.

- BS2_GetLog: Gets certain amount of logs.
- BS2_GetFilteredLog: Gets filtered logs.
- BS2_ClearLog: Deletes all logs.
- BS2_StartMonitoringLog: Starts Real-time log streaming.
- BS2_StartMonitoringLogEx: [+ V2.7.1] Starts Real-time log streaming including the temperature information.
- BS2_StopMonitoringLog: Stops Real-time log streaming.
- BS2_GetLogBlob: Gets certain amount of logs based on the event mask.
- BS2_GetFilteredLogSinceEventId: Gets filtered logs.
- BS2_GetImageLog : Gets image logs using event ID.
- BS2_GetLogSmallBlob: [+ 2.6.4] Gets certain amount of logs based on the event mask in an efficient way.
- BS2_GetLogSmallBlobEx: [+ 2.7.1] Gets certain amount of logs based on the event mask in an efficient way, including the temperature information.

Callback Function

OnLogReceived

Callback function that is called when receiving a new log from the device.

```
typedef void (*OnLogReceived)(uint32_t deviceId, BS2Event* log);
```

1. *deviceId*
Device ID

2. *log*
New log information structure

OnLogReceivedEx

[+ V2.7.1] Callback function that is called when receiving a new log from the device.
The temperature information is transferred as the 3rd parameter, and auditTemperature should be true in BS2FaceConfigExt.

```
typedef void (*OnLogReceivedEx)(uint32_t deviceId, BS2Event* log, uint32_t
temperature);
```

1. *deviceId*
Device ID

2. *log*
New log information structure

3. *temperature*
Temperature information

Structure

BS2Event

```
typedef struct {
    uint32_t id;
    uint32_t dateTime;
    uint32_t deviceID;
    union {
        char userID[BS2_USER_ID_SIZE];
        uint32_t uid;
        uint32_t doorID;
        uint32_t liftID;
        uint32_t zoneID;
        struct {
            uint32_t ioDeviceID;
            uint16_t port;
            int8_t value;
            uint8_t reserved[25];
        };
        struct {
            uint32_t zoneID;
            uint32_t doorID;
            uint32_t ioDeviceID;
            uint16_t port;
            uint8_t reserved[18];
        } alarm;
        struct {
            uint32_t zoneID;
            uint32_t doorID[4];
            uint8_t reserved[12];
        } interlock;
        struct {
            uint16_t relayPort;
            uint16_t inputPort;
            uint8_t reserved[28];
        } relayAction;
    };
```

```
    union {
        uint16_t code;
        struct {
            uint8_t subCode;
            uint8_t mainCode;
        };
    };
    uint8_t param;
#ifdef LESS_THAN_SDK_2_6_0
    BS2_BOOL image;         // Deprecated in V2.6.0
#else
    uint8_t image: 1;       // Support image and DST by bit division
    uint8_t isDST: 1;
    uint8_t half: 1;
    uint8_t hour: 4;
    uint8_t negative: 1;
#endif
} BS2Event;
```

1. *id*
Log record ID which automatically increases from 1 when the log is generated.

2. *dateTime*
The time when the log has been generated. It means the seconds past from UTC until the current time.

3. *deviceID*
ID of the device that generated the log.

4. *userID*
User ID related to log. When the value is 0, the log is not relevant to user.

5. *uid*
You can refer to doorID regarding door log, zoneID regarding zone log. If you do not know exactly then you can refer to uid.
Since uid, doorID, liftID, zoneID are declared as union, basically it means information such as unordered list.

6. *doorID*
ID of the door that generated the log.

7. *liftID*
ID of the lift that generated the log.

8. *zoneID*
ID of the zone that generated the log.

9. *ioDeviceID*
Door or Input device ID. When the value is 0, the log is not relevant to Door or Input device.

10. *port*
Port for ioDeviceID.

11. *value*
Port value for ioDeviceID and means below.
BS2_PORT_VALUE_UNKNOWN : -1
BS2_PORT_VALUE_OPEN : 0
BS2_PORT_VALUE_CLOSED : 1
BS2_PORT_VALUE_SUPERVISED_SHORT : 2
BS2_PORT_VALUE_SUPERVISED_OPEN : 3

12. *alarm.zoneID*
Zone ID that makes intrusion alarm zone alarms.

13. *alarm.doorID*
Door ID that makes intrusion alarm zone alarms.

14. *interlock.zoneID*
Zone ID that makes interlock zone alarms.

15. *interlock.doorID*
Door ID that makes interlock zone alarms.

16. *relayAction.relayPort*
Relay port identifier when IM-120 RelayAction takes place.

17. *relayAction.inputPort*
Relay port identifier when IM-120 RelayAction takes place.

18. *subCode*
Sub code value of log types. Use if the additional information is necessary.
^Category ^Event code ^Value ^Description ^ |Verify |BS2_SUB_EVENT_VERIFY_ID_PIN |0x01 |ID and PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_ID_FINGER |0x02 |ID and fingerprint verification success | |::: |BS2_SUB_EVENT_VERIFY_ID_FINGER_PIN |0x03 |ID, fingerprint, and PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_ID_FACE |0x04 |ID and face verification success | |::: |BS2_SUB_EVENT_VERIFY_ID_FACE_PIN |0x05 |ID, face, and PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_CARD |0x06 |Card verification success | |::: |BS2_SUB_EVENT_VERIFY_CARD_PIN |0x07 |Card and PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_CARD_FINGER |0x08 |Card and fingerprint verification success | |::: |BS2_SUB_EVENT_VERIFY_CARD_FINGER_PIN |0x09 |Card, fingerprint, and PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_CARD_FACE |0x0A |Card and face verification success | |::: |BS2_SUB_EVENT_VERIFY_CARD_FACE_PIN |0x0B |Card, PIN, and face verification success | |::: |BS2_SUB_EVENT_VERIFY_AOC |0x0C |AOC card verification success | |::: |BS2_SUB_EVENT_VERIFY_AOC_PIN |0x0D |AOC card and PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_AOC_FINGER |0x0E |AOC card and fingerprint verification success | |::: |BS2_SUB_EVENT_VERIFY_AOC_FINGER_PIN |0x0F |AOC card, fingerprint, PIN verification success | |::: |BS2_SUB_EVENT_VERIFY_MOBLIE_CARD |0x16 |Mobile card verification success (+V2.8) | |::: |BS2_SUB_EVENT_VERIFY_MOBILE_CARD_PIN |0x17 |Mobile card and PIN verification success (+V2.8) | |::: |BS2_SUB_EVENT_VERIFY_MOBILE_CARD_FINGER |0x18 |Mobile card and fingerprint verification success (+V2.8) | |::: |BS2_SUB_EVENT_VERIFY_MOBILE_CARD_FINGER_PIN

**|0x19 |Mobile card, fingerprint, PIN verification success (+V2.8) | |:::
|BS2_SUB_EVENT_VERIFY_MOBILE_CARD_FACE |0x1A |Mobile card and face verification success (+V2.8) | |::: |BS2_SUB_EVENT_VERIFY_MOBILE_CARD_FACE_PIN |0x1B |Mobile card, face, PIN verification success (+V2.8) | |:::
|BS2_SUB_EVENT_VERIFY_MOBILE_CARD_FACE_FINGER |0x20 |Mobile card, face, fingerprint verification success (+V2.8) | |::: |BS2_SUB_EVENT_VERIFY_MOBILE_CARD_FINGER_FACE |0x21 |Mobile card, fingerprint, face verification success (+V2.8) | |Identify
|BS2_SUB_EVENT_IDENTIFY_FINGER |0x01 |Fingerprint identification success | |:::
|BS2_SUB_EVENT_IDENTIFY_FINGER_PIN |0x02 |Fingerprint and PIN identification success | |::: |BS2_SUB_EVENT_IDENTIFY_FACE |0x03 |Face identification success | |:::
|BS2_SUB_EVENT_IDENTIFY_FACE_PIN |0x04 |Face and PIN identification success | |Auth
|BS2_SUB_EVENT_DUAL_AUTH_FAIL_TIMEOUT |0x01 |Dual authentication timeout | |:::
|BS2_SUB_EVENT_DUAL_AUTH_FAIL_ACCESS_GROUP |0x02 |Attempted the dual authentication with invalid user| |Credential |BS2_SUB_EVENT_CREDENTIAL_ID |0x01 |Invalid user ID| |::: |BS2_SUB_EVENT_CREDENTIAL_CARD |0x02 |Invalid card | |:::
|BS2_SUB_EVENT_CREDENTIAL_PIN |0x03 |Invalid PIN | |:::
|BS2_SUB_EVENT_CREDENTIAL_FINGER |0x04 |Invalid fingerprint | |:::
|BS2_SUB_EVENT_CREDENTIAL_FACE |0x05 |Invalid face | |:::
|BS2_SUB_EVENT_CREDENTIAL_AOC_PIN |0x06 |Invalid AOC PIN | |:::
|BS2_SUB_EVENT_CREDENTIAL_AOC_FINGER |0x07 |Invalid AOC fingerprint | |:::
|BS2_SUB_EVENT_CREDENTIAL_MOBILE_CARD |0x08 |Invalid mobile card (+V2.8) | |Auth
|BS2_SUB_EVENT_AUTH_FAIL_INVALID_AUTH_MODE |0x01 |Invalid authentication mode | |:::
|BS2_SUB_EVENT_AUTH_FAIL_INVALID_CREDENTIAL |0x02 |Non-registered authentication method| |::: |BS2_SUB_EVENT_AUTH_FAIL_TIMEOUT |0x03 |Authentication timeout | |Access
|BS2_SUB_EVENT_ACCESS_DENIED_ACCESS_GROUP |0x01 |Access was denied because the user has not been registered for the access group| |:::
|BS2_SUB_EVENT_ACCESS_DENIED_DISABLED |0x02 |Access was denied because the user is inactive| |::: |BS2_SUB_EVENT_ACCESS_DENIED_EXPIRED |0x03 |Access was denied because the user entry period was expired| |::: |BS2_SUB_EVENT_ACCESS_DENIED_ON_BLACKLIST |0x04 |Access was denied because the card is on the blacklist| |:::
|BS2_SUB_EVENT_ACCESS_DENIED_APB |0x05 |Access was denied because the user has violated the anti-passback rule| |::: |BS2_SUB_EVENT_ACCESS_DENIED_TIMED_APB |0x06 |Access was denied because the user tried to enter the timed anti-passback zone within the limited time frame| |::: |BS2_SUB_EVENT_ACCESS_DENIED_FORCED_LOCK |0x07 |Access was denied because the zone was forced to be locked| |APB
|BS2_SUB_EVENT_ZONE_HARD_APB |0x01 |Hard APB zone | |:::
|BS2_SUB_EVENT_ZONE_SOFT_APB |0x02 |Soft APB zone |** 19. *mainCode*
**Main code value of log types.
^Category ^Event code ^Value ^Description ^ |Auth |BS2_EVENT_VERIFY_SUCCESS |0x1000 |1:1 authentication success | |::: |BS2_EVENT_VERIFY_FAIL |0x1100 |1:1 authentication fail | |::: |BS2_EVENT_VERIFY_DURESS |0x1200 |1:1 duress authentication success | |::: |BS2_EVENT_IDENTIFY_SUCCESS |0x1300 |1:N authentication success | |:::
|BS2_EVENT_IDENTIFY_FAIL |0x1400 |1:N authentication fail | |:::
|BS2_EVENT_IDENTIFY_DURESS |0x1500 |1:N duress authentication success | |:::
|BS2_EVENT_DUAL_AUTH_SUCCESS |0x1600 |Dual authentication success | |:::
|BS2_EVENT_DUAL_AUTH_FAIL |0x1700 |Dual authentication fail | |:::
|BS2_EVENT_AUTH_FAILED |0x1800 |Attempted to authenticate with the non-registered credential| |::: |BS2_EVENT_ACCESS_DENIED |0x1900 |Invalid user attempted to authenticate or user violated the APB rule| |::: |BS2_EVENT_FAKE_FINGER_DETECTED |0x1A00 |Fake fingerprint detection | |User |BS2_EVENT_USER_ENROLL_SUCCESS |0x2000 |User enroll success | |::: |BS2_EVENT_USER_ENROLL_FAIL |0x2100 |User enroll fail | |:::**

**|BS2_EVENT_USER_UPDATE_SUCCESS |0x2200 |User update success | |:::
|BS2_EVENT_USER_UPDATE_FAIL |0x2300 |User update fail | |:::
|BS2_EVENT_USER_DELETE_SUCCESS |0x2400 |User delete success | |:::
|BS2_EVENT_USER_DELETE_FAIL |0x2500 |User delete fail | |:::
|BS2_EVENT_USER_DELETE_ALL_SUCCESS |0x2600 |Delete all user success | |:::
|BS2_EVENT_USER_ISSUE_AOC_SUCCESS |0x2700 |Authentication success with access card | |::: |BS2_EVENT_USER_DUPLICATE_CREDENTIAL |0x2800 |Duplicated credential(Card/Fingerprint/Face) detection| |Device |BS2_EVENT_DEVICE_SYSTEM_RESET |0x3000 |System reset | |::: |BS2_EVENT_DEVICE_SYSTEM_STARTED |0x3100 |System started | |::: |BS2_EVENT_DEVICE_TIME_SET |0x3200 |System time set | |:::
|BS2_EVENT_DEVICE_TIMEZONE_SET |0x3201 |Time zone chagned | |:::
|BS2_EVENT_DEVICE_DST_SET |0x3202 |DST setting changed | |:::
|BS2_EVENT_DEVICE_LINK_CONNECTED |0x3300 |LAN cable connected | |:::
|BS2_EVENT_DEVICE_LINK_DISCONNETED |0x3400 |LAN cable disconnected | |:::
|BS2_EVENT_DEVICE_DHCP_SUCCESS |0x3500 |IP address allocated by DHCP | |:::
|BS2_EVENT_DEVICE_ADMIN_MENU |0x3600 |Open administrator menu | |:::
|BS2_EVENT_DEVICE_UI_LOCKED |0x3700 |Screen locked | |:::
|BS2_EVENT_DEVICE_UI_UNLOCKED |0x3800 |Screen unlocked | |:::
|BS2_EVENT_DEVICE_COMM_LOCKED |0x3900 |RS485 communication locked | |:::
|BS2_EVENT_DEVICE_COMM_UNLOCKED |0x3A00 |RS485 communication unlocked | |:::
|BS2_EVENT_DEVICE_TCP_CONNECTED |0x3B00 |TCP connected | |:::
|BS2_EVENT_DEVICE_TCP_DISCONNECTED |0x3C00 |TCP disconnected | |:::
|BS2_EVENT_DEVICE_RS485_CONNECTED |0x3D00 |RS485 connected | |:::
|BS2_EVENT_DEVICE_RS485_DISCONNCTED |0x3E00 |RS485 disconnected | |:::
|BS2_EVENT_DEVICE_INPUT_DETECTED |0x3F00 |Input device detected | |:::
|BS2_EVENT_DEVICE_TAMPER_ON |0x4000 |Device or peripheral was removed| |:::
|BS2_EVENT_DEVICE_TAMPER_OFF |0x4100 |Device or peripheral was reconnected| |:::
|BS2_EVENT_DEVICE_EVENT_LOG_CLEARED |0x4200 |Log was deleted| |:::
|BS2_EVENT_DEVICE_FIRMWARE_UPGRADED |0x4300 |Firmware was updated| |:::
|BS2_EVENT_DEVICE_RESOURCE_UPGRADED |0x4400 |Resource was updated| |:::
|BS2_EVENT_DEVICE_CONFIG_RESET |0x4500 |System information was initialized(including network)| |::: |BS2_EVENT_DEVICE_DATABASE_RESET |0x4501 |Database was initialized | |:::
|BS2_EVENT_DEVICE_FACTORY_RESET |0x4502 |Factory default | |:::
|BS2_EVENT_DEVICE_CONFIG_RESET_EX |0x4503 |System information was initialized(without network) | |Supervised Input |BS2_EVENT_SUPERVISED_INPUT_SHORT |0x4600 |Supervised Input (Short circuit detection) | |:::
|BS2_EVENT_SUPERVISED_INPUT_OPEN |0x4700 |Supervised Input (Disconnection detection) | |Device-Ex |BS2_EVENT_DEVICE_AC_FAIL |0x4800 |AC Power failed | |:::
|BS2_EVENT_DEVICE_AC_SUCCESS |0x4900 |AC Power succeeded | |Door
|BS2_EVENT_DOOR_UNLOCKED |0x5000 |Door unlocked | |::: |BS2_EVENT_DOOR_LOCKED |0x5100 |Door locked | |::: |BS2_EVENT_DOOR_OPENED |0x5200 |Door opened | |:::
|BS2_EVENT_DOOR_CLOSED |0x5300 |Door closed | |::: |BS2_EVENT_DOOR_FORCED_OPEN |0x5400 |Door forced open | |::: |BS2_EVENT_DOOR_HELD_OPEN |0x5500 |Door held open | |::: |BS2_EVENT_DOOR_FORCED_OPEN_ALARM |0x5600 |Door-forced-to-open alarm has started| |::: |BS2_EVENT_DOOR_FORCED_OPEN_ALARM_CLEAR |0x5700 |Door-forced-to-open alarm was released| |::: |BS2_EVENT_DOOR_HELD_OPEN_ALARM |0x5800 |Door-held-open alarm has started| |::: |BS2_EVENT_DOOR_HELD_OPEN_ALARM_CLEAR |0x5900 |Door-held-open alarm was released| |::: |BS2_EVENT_DOOR_APB_ALARM |0x5A00 |Door-level anti-passback alarm has started| |::: |BS2_EVENT_DOOR_APB_ALARM_CLEAR |0x5B00 |Door-level anti-passback alarm was released| |Zone |BS2_EVENT_ZONE_APB_VIOLATION |0x6000 |Zone-level anti-passback rule has been violated| |::: |BS2_EVENT_ZONE_APB_ALARM**

|0x6100 |Zone-level anti-passback alarm has started| |::: |BS2_EVENT_ZONE_APB_ALARM_CLEAR |0x6200 |Zone-level anti-passback alarm was released| |::: |BS2_EVENT_ZONE_TIMED_APB_VIOLATION |0x6300 |Timed anti-passback rule has been violated| |::: |BS2_EVENT_ZONE_TIMED_APB_ALARM |0x6400 |Timed anti-passback alarm has started| |::: |BS2_EVENT_ZONE_TIMED_APB_ALARM_CLEAR |0x6500 |Timed anti-passback alarm was released| |::: |BS2_EVENT_ZONE_FIRE_ALARM_INPUT |0x6600 |Fire alarm input was detected| |::: |BS2_EVENT_ZONE_FIRE_ALARM |0x6700 |Fire alarm has started| |::: |BS2_EVENT_ZONE_FIRE_ALARM_CLEAR |0x6800 |Fire alarm was released| |::: |BS2_EVENT_ZONE_FORCED_LOCK_START |0x6900 |Door-forced-locked schedule has started| |::: |BS2_EVENT_ZONE_FORCED_LOCK_END |0x6A00 |Door-forced-locked schedule has ended| |::: |BS2_EVENT_ZONE_FORCED_UNLOCK_START |0x6B00 |Door-forced-unlocked schedule has started| |::: |BS2_EVENT_ZONE_FORCED_UNLOCK_END |0x6C00 |Door-forced-unlocked schedule has ended| |::: |BS2_EVENT_ZONE_SCHEDULED_UNLOCK_END |0x6D00 |Scheduled unlock ended | |::: |BS2_EVENT_ZONE_SCHEDULED_LOCK_ALARM |0x6E00 |Scheduled lock alarm zone start | |::: |BS2_EVENT_ZONE_SCHEDULED_LOCK_ALARM_CLEAR |0x6F00 |Scheduled lock alarm zone clear | | |RelayAction |BS2_EVENT_RELAY_ACTION_ON |0xC300 |RelayAction Switch-ON | |::: |BS2_EVENT_RELAY_ACTION_OFF |0xC400 |RelayAction Switch-OFF | |::: |BS2_EVENT_RELAY_ACTION_KEEP |0xC500 |RelayAction KEEP SIGNAL | 20.*param*

It is used only when extra information on the device is needed. Usually, a time and attendance code, a port number of the door or input device is stored in the *param* argument. When a time and attendance code is stored, refer to the following values:

Device Type	T&A Code	Mapped Key	Value
BioStation 2	BS2_TNA_UNSPECIFIED	(N/A)	0
	BS2_TNA_KEY_1	F1	1
	BS2_TNA_KEY_2	F2	2
	BS2_TNA_KEY_3	F3	3
	BS2_TNA_KEY_4	F4	4
	BS2_TNA_KEY_5	1	5
	BS2_TNA_KEY_6	2	6
	BS2_TNA_KEY_7	3	7
	BS2_TNA_KEY_8	4	8
	BS2_TNA_KEY_9	5	9
	BS2_TNA_KEY_10	6	10
	BS2_TNA_KEY_11	7	11
	BS2_TNA_KEY_12	8	12
	BS2_TNA_KEY_13	9	13
	BS2_TNA_KEY_14	Call	14
	BS2_TNA_KEY_15	0	15
	BS2_TNA_KEY_16	Esc	16

[+ 2.6.3] Additional feature of param

Only if the event code is relevant to the user.

If you add, modify or delete users directly from the device, param is 1, 0 if done through BioStar.

For example, if param was 1 with the event BS2_EVENT_USER_ENROLL_SUCCESS, it means the user was added directly on the device.

This feature is supported from the firmware version below.

Device Type	Supported Ver.
BioStation 2	V1.7.0
BioStation A2	V1.6.0
CoreStation 40	V1.2.0
BioEntry P2	V1.2.0
BioStation L2	V1.4.0
BioLite N2	V1.1.0
BioEntry W2	V1.3.0
FaceStation 2	V1.2.0

21. *image***
Prior to SDK V2.6.0, it used the whole 1 byte and means below:
- Whether the image was included when the event occurred (true / false).

Since SDK V2.6.0, 1 byte has been changed to provide the following information by bit unit.
- Whether or not image is included. - Whether DST is applied

Category	Bit	Parameter	Description
Prior to SDK 2.6.0	8	image	Used in case image is included when an event occurs.
Since SDK 2.6.0	1	image	Used in case image is included when an event occurs.
	1	isDST	Whether the current log has been applied to DST
	1	half	Whether DST is covered in 30-minute increments. 0 is 0 minutes, 1 is 30 minutes.
	4	hour	Time. 1 to 12 o'clock
	1	negative	0 is +, 1 is -

BS2EventBlob

```
typedef struct {
    uint16_t eventMask;
    uint32_t id;
    BS2EventExtInfo info;
    union
    {
    BS2_USER_ID userID;                     // valid if eventMask has
BS2_EVENT_MASK_USER_ID
    uint8_t cardID[BS2_CARD_DATA_SIZE];     // valid if eventMask has
BS2_EVENT_MASK_CARD_ID
    BS2_DOOR_ID doorID;                     // valid if eventMask has
BS2_EVENT_MASK_DOOR_ID
    BS2_ZONE_ID zoneID;                     // valid if eventMask has
BS2_EVENT_MASK_ZONE_ID
    BS2EventExtIoDevice ioDevice;           // valid if eventMask has
BS2_EVENT_MASK_IODEVICE
    };
    uint8_t tnaKey;
```

```
    uint32_t jobCode;
    uint16_t imageSize;
    uint8_t image[BS2_EVENT_MAX_IMAGE_SIZE];
    uint8_t reserved;
} BS2EventBlob;
```

1. *eventMask*
Event mask value. Logs will be retrieved based on the mask value such as user, card, door, or zone.

Value	Description
0x0000	None
0x0001	BS2EventExtInfo Structure
0x0002	User ID
0x0004	Card ID
0x0008	Door ID
0x0010	Zone ID
0x0020	BS2EventExtIoDevice Structure
0x0040	TNA Key
0x0080	Job Code
0x0100	Image
0x0200	Temperature
0xFFFF	ALL

2. *id*
Log record ID which automatically increases from 1 when the log is generated.

3. *info*
BS2EventExtInfo structure information.

4. *userID*
User ID related to log. When the value is 0, the log is not relevant to user.

5. *cardID*
Card ID related to log. When the value is 0, the log is not relevant to card.
The device records card ID only for failure auth, it returns user ID when successful auth regardless of eventMask value.

6. *doorID*
Door ID related to log. When the value is 0, the log is not relevant to door.

7. *zoneID*
Zone ID related to log. When the value is 0, the log is not relevant to zone.

8. *ioDevice*
Door or input device ID related to log. When the value is 0, the log is not relevant to door or input.
Refer to BS2EventExtIoDevice structure.

9. *tnaKey*
The T&A key that has been used for the authentication. When the value is 0, the log is not relevant to T&A key.

10. *jobCode*
The job code that has been used for the authentication. When the value is 0, the log is not relevant to job code.

11. *imageSize*
Size of the image when there is an image log.

12. *image*
Data of the image.

13. *reserved*
Reserved space.

BS2EventExtInfo

```
typedef struct {
    uint32_t dateTime;
    uint32_t deviceID;
    union {                                         ///< 2 bytes
        BS2_EVENT_CODE code;
        struct {
            uint8_t subCode;
            uint8_t mainCode;
        };
    };
    uint8_t reserved[2];
} BS2EventExtInfo;
```

1. *dateTime*
The time when the log has been generated. It means the seconds past from UTC until the current time.

2. *deviceID*
ID of the device that generated the log.

3. *subCode*
Sub code value of log types. Use if the additional information is necessary.

4. *mainCode*
Main code value of log types.

5. *reserved*
Reserved space.

BS2EventExtIoDevice

```
typedef struct {
    uint32_t ioDeviceID;
    uint16_t port;
    uint8_t value;
    uint8_t reserved[1];
} BS2EventExtInfo;
```

1. *ioDeviceID*
Door or input device ID related to log. When the value is 0, the log is not relevant to door or input.

2. *port*
Input port number.

3. *value*
Status of the input port.

Value	Description
-1	Unknown
0	Open
1	Closed
2	Supervised Short
3	Supervised Open

4. *reserved*
Reserved space.

BS2EventSmallBlob

```
typedef struct {
    uint16_t eventMask;
    uint32_t id;
    BS2EventExtInfo info;
    union
    {
    BS2_USER_ID userID;                     // valid if eventMask has
BS2_EVENT_MASK_USER_ID
    uint8_t cardID[BS2_CARD_DATA_SIZE];     // valid if eventMask has
BS2_EVENT_MASK_CARD_ID
    BS2_DOOR_ID doorID;                     // valid if eventMask has
BS2_EVENT_MASK_DOOR_ID
    BS2_ZONE_ID zoneID;                     // valid if eventMask has
BS2_EVENT_MASK_ZONE_ID
    BS2EventExtIoDevice ioDevice;           // valid if eventMask has
```

```
BS2_EVENT_MASK_IODEVICE
    };
    uint8_t tnaKey;
    uint32_t jobCode;
    uint16_t imageSize;
    uint8_t* imageObj;                          // valid if eventMask has
BS2_EVENT_MASK_IMAGE
    uint8_t reserved;
} BS2EventSmallBlob;
```

1. *eventMask*
Event mask value. Logs will be retrieved based on the mask value such as user, card, door, or zone.

Value	Description
0x0000	None
0x0001	BS2EventExtInfo Structure
0x0002	User ID
0x0004	Card ID
0x0008	Door ID
0x0010	Zone ID
0x0020	BS2EventExtIoDevice Structure
0x0040	TNA Key
0x0080	Job Code
0x0100	Image
0x0200	Temperature
0xFFFF	ALL

2. *id*
Log record ID which automatically increases from 1 when the log is generated.

3. *info*
BS2EventExtInfo structure information.

4. *userID*
User ID related to log. When the value is 0, the log is not relevant to user.

5. *cardID*
Card ID related to log. When the value is 0, the log is not relevant to card.
The device records card ID only for failure auth, it returns user ID when successful auth regardless of eventMask value.

6. *doorID*
Door ID related to log. When the value is 0, the log is not relevant to door.

7. *zoneID*
Zone ID related to log. When the value is 0, the log is not relevant to zone.

8. *ioDevice*
Door or input device ID related to log. When the value is 0, the log is not relevant to door or input. Refer to BS2EventExtIoDevice structure.

9. *tnaKey*
The T&A key that has been used for the authentication. When the value is 0, the log is not relevant to T&A key.

10. *jobCode*
The job code that has been used for the authentication. When the value is 0, the log is not relevant to job code.

11. *imageSize*
Size of the image when there is an image log.

12. *imageObj*
Data of the image.

13. *reserved*
Reserved space.

BS2EventSmallBlobEx

```
typedef struct {
    uint16_t eventMask;
    uint32_t id;
    BS2EventExtInfo info;                       // valid if eventMask has
BS2_EVENT_MASK_INFO
    union
    {
        BS2_USER_ID userID;                     // valid if eventMask has
BS2_EVENT_MASK_USER_ID
        uint8_t cardID[BS2_CARD_DATA_SIZE];     // valid if eventMask has
BS2_EVENT_MASK_CARD_ID
        BS2_DOOR_ID doorID;                     // valid if eventMask has
BS2_EVENT_MASK_DOOR_ID
        BS2_ZONE_ID zoneID;                     // valid if eventMask has
BS2_EVENT_MASK_ZONE_ID
        BS2EventExtIoDevice ioDevice;           // valid if eventMask has
BS2_EVENT_MASK_IODEVICE
    };
    uint8_t tnaKey;                             // valid if eventMask has
BS2_EVENT_MASK_TNA_KEY
    uint32_t jobCode;                           // valid if eventMask has
BS2_EVENT_MASK_JOB_CODE
    uint16_t imageSize;                         // valid if eventMask has
BS2_EVENT_MASK_IMAGE
    uint8_t* imageObj;                          // valid if eventMask has
BS2_EVENT_MASK_IMAGE
```

```
    uint8_t reserved;
    uint32_t temperature;                       // valid if eventMask has
BS2_EVENT_MASK_TEMPERATURE
} BS2EventSmallBlobEx;
```

1. *eventMask*
Event mask value. Logs will be retrieved based on the mask value such as user, card, door, or zone.

Value	Description
0x0000	None
0x0001	BS2EventExtInfo Structure
0x0002	User ID
0x0004	Card ID
0x0008	Door ID
0x0010	Zone ID
0x0020	BS2EventExtIoDevice Structure
0x0040	TNA Key
0x0080	Job Code
0x0100	Image
0x0200	Temperature
0xFFFF	ALL

2. *id*
Log record ID which automatically increases from 1 when the log is generated.

3. *info*
BS2EventExtInfo structure information.

4. *userID*
User ID related to log. When the value is 0, the log is not relevant to user.

5. *cardID*
Card ID related to log. When the value is 0, the log is not relevant to card.
The device records card ID only for failure auth, it returns user ID when successful auth regardless of eventMask value.

6. *doorID*
Door ID related to log. When the value is 0, the log is not relevant to door.

7. *zoneID*
Zone ID related to log. When the value is 0, the log is not relevant to zone.

8. *ioDevice*
Door or input device ID related to log. When the value is 0, the log is not relevant to door or input.
Refer to BS2EventExtIoDevice structure.

9. *tnaKey*

The T&A key that has been used for the authentication. When the value is 0, the log is not relevant to T&A key.

10. *jobCode*
The job code that has been used for the authentication. When the value is 0, the log is not relevant to job code.

11. *imageSize*
Size of the image when there is an image log.

12. *image*
Data of the image.

13. *temperature*
Temperature information is set when there's a thermal camera installed and setting to record event logs. Refer to auditTemperature

14. *reserved*
Reserved.