# 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_StopMonitoringLog: Stops Real-time log streaming.
- BS2_GetLogBlob: Gets certain amount of logs based on the event mask.
- BS2_GetFilteredLogSinceEventId: Gets filtered logs.

## Callback Function

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

**1. *OnLogReceived***
Callback function that is called when a new log is received.

## Structure

### BS2Event

```
typedef struct {
    uint32_t id;
    uint32_t dateTime;
    uint32_t deviceID;
    union {
        char userID[BS2_USER_ID_SIZE];
        uint32_t ioDeviceID;
    };
    union {
        uint16_t code;
        struct {
            uint8_t mainCode;
            uint8_t subCode;
        };
    };
    uint8_t param;
    uint8_t image;
} 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. *ioDeviceID***
Door or Input device ID. When the value is 0, the log is not relevant to Door or Input device.

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

| Category | Event code | Value | Description |
|---|---|---|---|
| Auth | BS2_EVENT_VERIFY_SUCCESS | 0x10 | 1:1 authentication success |
| | BS2_EVENT_VERIFY_FAIL | 0x11 | 1:1 authentication fail |
| | BS2_EVENT_VERIFY_DURESS | 0x12 | 1:1 duress authentication success |
| | BS2_EVENT_IDENTIFY_SUCCESS | 0x13 | 1:N authentication success |
| | BS2_EVENT_IDENTIFY_FAIL | 0x14 | 1:N authentication fail |
| | BS2_EVENT_IDENTIFY_DURESS | 0x15 | 1:N duress authentication success |
| | BS2_EVENT_DUAL_AUTH_SUCCESS | 0x16 | Dual authentication success |
| | BS2_EVENT_DUAL_AUTH_FAIL | 0x17 | Dual authentication fail |
| | BS2_EVENT_AUTH_FAILED | 0x18 | Attempted to authenticate with the non-registered credential |
| | BS2_EVENT_ACCESS_DENIED | 0x19 | Invalid user attempted to authenticate or user violated the APB rule |
| User | BS2_EVENT_USER_ENROLL_SUCCESS | 0x20 | User enroll success |
| | BS2_EVENT_USER_ENROLL_FAIL | 0x21 | User enroll fail |
| | BS2_EVENT_USER_UPDATE_SUCCESS | 0x22 | User update success |
| | BS2_EVENT_USER_UPDATE_FAIL | 0x23 | User update fail |
| | BS2_EVENT_USER_DELETE_SUCCESS | 0x24 | User delete success |
| | BS2_EVENT_USER_DELETE_FAIL | 0x25 | User delete fail |
| | BS2_EVENT_USER_DELETE_ALL_SUCCESS | 0x26 | Delete all user success |
| | BS2_EVENT_USER_ISSUE_AOC_SUCCESS | 0x26 | Authentication success with access card |

| Category | Event code | Value | Description |
|---|---|---|---|
| Device | BS2_EVENT_DEVICE_SYSTEM_RESET | 0x30 | System reset |
| | BS2_EVENT_DEVICE_SYSTEM_STARTED | 0x31 | System started |
| | BS2_EVENT_DEVICE_TIME_SET | 0x32 | System time set |
| | BS2_EVENT_DEVICE_LINK_CONNECTED | 0x33 | LAN cable connected |
| | BS2_EVENT_DEVICE_LINK_DISCONNETED | 0x34 | LAN cable disconnected |
| | BS2_EVENT_DEVICE_DHCP_SUCCESS | 0x35 | IP address allocated by DHCP |
| | BS2_EVENT_DEVICE_ADMIN_MENU | 0x36 | Open administrator menu |
| | BS2_EVENT_DEVICE_UI_LOCKED | 0x37 | Screen locked |
| | BS2_EVENT_DEVICE_UI_UNLOCKED | 0x38 | Screen unlocked |
| | BS2_EVENT_DEVICE_COMM_LOCKED | 0x39 | RS485 communication locked |
| | BS2_EVENT_DEVICE_COMM_UNLOCKED | 0x3A | RS485 communication unlocked |
| | BS2_EVENT_DEVICE_TCP_CONNECTED | 0x3B | TCP connected |
| | BS2_EVENT_DEVICE_TCP_DISCONNECTED | 0x3C | TCP disconnected |
| | BS2_EVENT_DEVICE_RS485_CONNECTED | 0x3D | RS485 connected |
| | BS2_EVENT_DEVICE_RS485_DISCONNCTED | 0x3E | RS485 disconnected |
| | BS2_EVENT_DEVICE_INPUT_DETECTED | 0x3F | Input device detected |
| | BS2_EVENT_DEVICE_TAMPER_ON | 0x40 | Device or peripheral was removed |
| | BS2_EVENT_DEVICE_TAMPER_OFF | 0x41 | Device or peripheral was reconnected |
| | BS2_EVENT_DEVICE_EVENT_LOG_CLEARED | 0x42 | Log was deleted |
| | BS2_EVENT_DEVICE_FIRMWARE_UPGRADED | 0x43 | Firmware was updated |
| | BS2_EVENT_DEVICE_RESOURCE_UPGRADED | 0x44 | Resource was updated |
| | BS2_EVENT_DEVICE_CONFIG_RESET | 0x45 | System information was initialized |
| Door | BS2_EVENT_DOOR_UNLOCKED | 0x50 | Door unlocked |
| | BS2_EVENT_DOOR_LOCKED | 0x51 | Door locked |
| | BS2_EVENT_DOOR_OPENED | 0x52 | Door opened |
| | BS2_EVENT_DOOR_CLOSED | 0x53 | Door closed |
| | BS2_EVENT_DOOR_FORCED_OPEN | 0x54 | Door forced open |
| | BS2_EVENT_DOOR_HELD_OPEN | 0x55 | Door held open |
| | BS2_EVENT_DOOR_FORCED_OPEN_ALARM | 0x56 | Door-forced-to-open alarm has started |
| | BS2_EVENT_DOOR_FORCED_OPEN_ALARM_CLEAR | 0x57 | Door-forced-to-open alarm was released |
| | BS2_EVENT_DOOR_HELD_OPEN_ALARM | 0x58 | Door-held-open alarm has started |
| | BS2_EVENT_DOOR_HELD_OPEN_ALARM_CLEAR | 0x59 | Door-held-open alarm was released |
| | BS2_EVENT_DOOR_APB_ALARM | 0x5A | Door-level anti-passback alarm has started |
| | BS2_EVENT_DOOR_APB_ALARM_CLEAR | 0x5B | Door-level anti-passback alarm was released |

| Category | Event code | Value | Description |
|---|---|---|---|
| Zone | BS2_EVENT_ZONE_APB_VIOLATION | 0x60 | Zone-level anti-passback rule has been violated |
| | BS2_EVENT_ZONE_APB_ALARM | 0x61 | Zone-level anti-passback alarm has started |
| | BS2_EVENT_ZONE_APB_ALARM_CLEAR | 0x62 | Zone-level anti-passback alarm was released |
| | BS2_EVENT_ZONE_TIMED_APB_VIOLATION | 0x63 | Timed anti-passback rule has been violated |
| | BS2_EVENT_ZONE_TIMED_APB_ALARM | 0x64 | Timed anti-passback alarm has started |
| | BS2_EVENT_ZONE_TIMED_APB_ALARM_CLEAR | 0x65 | Timed anti-passback alarm was released |
| | BS2_EVENT_ZONE_FIRE_ALARM_INPUT | 0x66 | Fire alarm input was detected |
| | BS2_EVENT_ZONE_FIRE_ALARM | 0x67 | Fire alarm has started |
| | BS2_EVENT_ZONE_FIRE_ALARM_CLEAR | 0x68 | Fire alarm was released |
| | BS2_EVENT_ZONE_FORCED_LOCK_START | 0x69 | Door-forced-locked schedule has started |
| | BS2_EVENT_ZONE_FORCED_LOCK_END | 0x6A | Door-forced-locked schedule has ended |
| | BS2_EVENT_ZONE_FORCED_UNLOCK_START | 0x6B | Door-forced-unlocked schedule has started |
| | BS2_EVENT_ZONE_FORCED_UNLOCK_END | 0x6C | Door-forced-unlocked schedule has ended |

### 7. *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 | Smart card verification success |
| | BS2_SUB_EVENT_VERIFY_CARD_PIN | 0x07 | Smart card and PIN verification success |
| | BS2_SUB_EVENT_VERIFY_CARD_FINGER | 0x08 | Smart card and fingerprint verification success |
| | BS2_SUB_EVENT_VERIFY_CARD_FINGER_PIN | 0x09 | Smart card, fingerprint, and PIN verification success |
| | BS2_SUB_EVENT_VERIFY_CARD_FACE | 0x0A | Smart card and face verification success |
| | BS2_SUB_EVENT_VERIFY_CARD_FACE_PIN | 0x0B | Smart 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 |
| 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 Smart 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 |
| 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 |

| Category | Event code | Value | Description |
|---|---|---|---|
| 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 |

### 8. *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 |

### 9. *image*

It is used when the event includes image information.

## 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 |
|---|---|
| 0 | None |
| 1 | BS2EventExtInfo structure |
| 2 | User ID |
| 4 | Card ID |
| 8 | Door ID |
| 16 | Zone ID |
| 32 | BS2EventExtIoDevice structure |
| 64 | Door ID |
| 128 | Zone ID |
| 256 | TNA Key |
| 512 | Job Code |
| 1024 | Image |
| 65535 | ALL |

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

logs are all deleted.

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.

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.