Log Management API

API .

- [BS2_GetLog](): .
- [BS2_GetFilteredLog](): .
- [BS2_ClearLog](): .
- [BS2_StartMonitoringLog](): .
- [BS2_StopMonitoringLog](): .
- [BS2_GetLogBlob](): EventMask .
- [BS2_GetFilteredLogSinceEventId](): .

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

1. *OnLogReceived*

.

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 subCode;
            uint8_t mainCode;
        };
    };
    uint8_t param;
    uint8_t image;
} BS2Event;

typedef struct {
```

```
    union {
        BS2_USER_ID        userID;                ///< 32 bytes
        uint32_t uid; (        ID)
        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;
    };
    union {                                       ///< 2 bytes
        uint16_t code;
        struct {
            uint8_t    subCode;
            uint8_t    mainCode;
        };
    };
    uint8_t                param;                 ///< 1 byte : tnaKey,
floorIndex, alarmFlags
#ifdef DST_SUPPORTED      // BSPP-7
    uint8_t image: 1;
    uint8_t isDST: 1;
    uint8_t half: 1;
    uint8_t hour: 4;
    uint8_t negative: 1;
#else
    BS2_BOOL        image;                        ///< 1 byte
#endif
} BS2Event;
```

1. *id*

순차적으로 증가하는 이벤트의 1 번째 식별자이며 순차적으로 증가합니다.

2. *dateTime*

이벤트가 발생한 시간이며, UTC 기준으로 초단위의 값으로 (sec) 저장됩니다.

3. *deviceID*

.

4. *userID*

O .

5. *uid*

32bit .

6. *doorID*

O .

7. *liftID*

O .

8. *zoneID*

O .

5. *ioDeviceID*

Door Input Door Input O .

6. *subCode*

. .

Verify	BS2_SUB_EVENT_VERIFY_ID_PIN	0x01	PIN
	BS2_SUB_EVENT_VERIFY_ID_FINGER	0x02	
	BS2_SUB_EVENT_VERIFY_ID_FINGER_PIN	0x03	PIN
	BS2_SUB_EVENT_VERIFY_ID_FACE	0x04	
	BS2_SUB_EVENT_VERIFY_ID_FACE_PIN	0x05	PIN
	BS2_SUB_EVENT_VERIFY_CARD	0x06	
	BS2_SUB_EVENT_VERIFY_CARD_PIN	0x07	PIN
	BS2_SUB_EVENT_VERIFY_CARD_FINGER	0x08	
	BS2_SUB_EVENT_VERIFY_CARD_FINGER_PIN	0x09	, , PIN
	BS2_SUB_EVENT_VERIFY_CARD_FACE	0x0A	
	BS2_SUB_EVENT_VERIFY_CARD_FACE_PIN	0x0B	, , PIN
	BS2_SUB_EVENT_VERIFY_AOC	0x0C	AOC
	BS2_SUB_EVENT_VERIFY_AOC_PIN	0x0D	AOC PIN
	BS2_SUB_EVENT_VERIFY_AOC_FINGER	0x0E	AOC
	BS2_SUB_EVENT_VERIFY_AOC_FINGER_PIN	0x0F	AOC , , PIN

Identify	BS2_SUB_EVENT_IDENTIFY_FINGER	0x01	
	BS2_SUB_EVENT_IDENTIFY_FINGER_PIN	0x02	PIN
	BS2_SUB_EVENT_IDENTIFY_FACE	0x03	
	BS2_SUB_EVENT_IDENTIFY_FACE_PIN	0x04	PIN
Auth	BS2_SUB_EVENT_DUAL_AUTH_FAIL_TIMEOUT	0x01	(2)
	BS2_SUB_EVENT_DUAL_AUTH_FAIL_ACCESS_GROUP	0x02	2
Credential	BS2_SUB_EVENT_CREDENTIAL_ID	0x01	
	BS2_SUB_EVENT_CREDENTIAL_CARD	0x02	
	BS2_SUB_EVENT_CREDENTIAL_PIN	0x03	PIN
	BS2_SUB_EVENT_CREDENTIAL_FINGER	0x04	
	BS2_SUB_EVENT_CREDENTIAL_FACE	0x05	
	BS2_SUB_EVENT_CREDENTIAL_AOC_PIN	0x06	AOC PIN
	BS2_SUB_EVENT_CREDENTIAL_AOC_FINGER	0x07	AOC
Auth	BS2_SUB_EVENT_AUTH_FAIL_INVALID_AUTH_MODE	0x01	
	BS2_SUB_EVENT_AUTH_FAIL_INVALID_CREDENTIAL	0x02	
	BS2_SUB_EVENT_AUTH_FAIL_TIMEOUT	0x03	
Access	BS2_SUB_EVENT_ACCESS_DENIED_ACCESS_GROUP	0x01	
	BS2_SUB_EVENT_ACCESS_DENIED_DISABLED	0x02	
	BS2_SUB_EVENT_ACCESS_DENIED_EXPIRED	0x03	
	BS2_SUB_EVENT_ACCESS_DENIED_ON_BLACKLIST	0x04	
	BS2_SUB_EVENT_ACCESS_DENIED_APB	0x05	APB
	BS2_SUB_EVENT_ACCESS_DENIED_TIMED_APB	0x06	Timed APB
	BS2_SUB_EVENT_ACCESS_DENIED_FORCED_LOCK	0x07	
APB	BS2_SUB_EVENT_ZONE_HARD_APB	0x01	APB
	BS2_SUB_EVENT_ZONE_SOFT_APB	0x02	APB

7. *mainCode*

.

Auth	BS2_EVENT_VERIFY_SUCCESS	0x1000	1:1
	BS2_EVENT_VERIFY_FAIL	0x1100	1:1
	BS2_EVENT_VERIFY_DURESS	0x1200	1:1
	BS2_EVENT_IDENTIFY_SUCCESS	0x1300	1:N
	BS2_EVENT_IDENTIFY_FAIL	0x1400	1:N
	BS2_EVENT_IDENTIFY_DURESS	0x1500	1:N
	BS2_EVENT_DUAL_AUTH_SUCCESS	0x1600	(2)
	BS2_EVENT_DUAL_AUTH_FAIL	0x1700	(2)
	BS2_EVENT_AUTH_FAILED	0x1800	
	BS2_EVENT_ACCESS_DENIED	0x1900	APB
	BS2_EVENT_FAKE_FINGER_DETECTED	0x1A00	
User	BS2_EVENT_USER_ENROLL_SUCCESS	0x2000	
	BS2_EVENT_USER_ENROLL_FAIL	0x2100	
	BS2_EVENT_USER_UPDATE_SUCCESS	0x2200	
	BS2_EVENT_USER_UPDATE_FAIL	0x2300	
	BS2_EVENT_USER_DELETE_SUCCESS	0x2400	
	BS2_EVENT_USER_DELETE_FAIL	0x2500	
	BS2_EVENT_USER_DELETE_ALL_SUCCESS	0x2600	
	BS2_EVENT_USER_ISSUE_AOC_SUCCESS	0x2700	Access card
Device	BS2_EVENT_DEVICE_SYSTEM_RESET	0x3000	
	BS2_EVENT_DEVICE_SYSTEM_STARTED	0x3100	
	BS2_EVENT_DEVICE_TIME_SET	0x3200	
	BS2_EVENT_DEVICE_LINK_CONNECTED	0x3300	LAN
	BS2_EVENT_DEVICE_LINK_DISCONNETED	0x3400	LAN
	BS2_EVENT_DEVICE_DHCP_SUCCESS	0x3500	DHCP IP
	BS2_EVENT_DEVICE_ADMIN_MENU	0x3600	
	BS2_EVENT_DEVICE_UI_LOCKED	0x3700	
	BS2_EVENT_DEVICE_UI_UNLOCKED	0x3800	
	BS2_EVENT_DEVICE_COMM_LOCKED	0x3900	RS485
	BS2_EVENT_DEVICE_COMM_UNLOCKED	0x3A00	RS485
	BS2_EVENT_DEVICE_TCP_CONNECTED	0x3B00	TCP
	BS2_EVENT_DEVICE_TCP_DISCONNECTED	0x3C00	TCP
	BS2_EVENT_DEVICE_RS485_CONNECTED	0x3D00	RS485
	BS2_EVENT_DEVICE_RS485_DISCONNCTED	0x3E00	RS485
	BS2_EVENT_DEVICE_INPUT_DETECTED	0x3F00	
	BS2_EVENT_DEVICE_TAMPER_ON	0x4000	
	BS2_EVENT_DEVICE_TAMPER_OFF	0x4100	
	BS2_EVENT_DEVICE_EVENT_LOG_CLEARED	0x4200	
	BS2_EVENT_DEVICE_FIRMWARE_UPGRADED	0x4300	
	BS2_EVENT_DEVICE_RESOURCE_UPGRADED	0x4400	
	BS2_EVENT_DEVICE_CONFIG_RESET	0x4500	
	BS2_EVENT_DEVICE_DATABASE_RESET	0x4501	
	BS2_EVENT_DEVICE_FACTORY_RESET	0x4502	
	BS2_EVENT_DEVICE_CONFIG_RESET_EX	0x4503	
Supervised Input	BS2_EVENT_SUPERVISED_INPUT_SHORT	0x4600	Supervised Input ( )
	BS2_EVENT_SUPERVISED_INPUT_OPEN	0x4700	Supervised Input ( )
Device-Ex	BS2_EVENT_DEVICE_AC_FAIL	0x4800	AC Power
	BS2_EVENT_DEVICE_AC_SUCCESS	0x4900	AC Power

Door	BS2_EVENT_DOOR_UNLOCKED	0x5000	
	BS2_EVENT_DOOR_LOCKED	0x5100	
	BS2_EVENT_DOOR_OPENED	0x5200	
	BS2_EVENT_DOOR_CLOSED	0x5300	
	BS2_EVENT_DOOR_FORCED_OPEN	0x5400	
	BS2_EVENT_DOOR_HELD_OPEN	0x5500	
	BS2_EVENT_DOOR_FORCED_OPEN_ALARM	0x5600	BS2_EVENT_DOOR_FORCED_OPEN
	BS2_EVENT_DOOR_FORCED_OPEN_ALARM_CLEAR	0x5700	BS2_EVENT_DOOR_FORCED_OPEN
	BS2_EVENT_DOOR_HELD_OPEN_ALARM	0x5800	BS2_EVENT_DOOR_HELD_OPEN
	BS2_EVENT_DOOR_HELD_OPEN_ALARM_CLEAR	0x5900	BS2_EVENT_DOOR_HELD_OPEN
	BS2_EVENT_DOOR_APB_ALARM	0x5A00	APB
	BS2_EVENT_DOOR_APB_ALARM_CLEAR	0x5B00	APB
	BS2_EVENT_DOOR_RELEASE	0x5C00	
	BS2_EVENT_DOOR_LOCK	0x5D00	
	BS2_EVENT_DOOR_UNLOCK	0x5E00	
Zone	BS2_EVENT_ZONE_APB_VIOLATION	0x6000	APB
	BS2_EVENT_ZONE_APB_ALARM	0x6100	BS2_EVENT_ZONE_APB_VIOLATION
	BS2_EVENT_ZONE_APB_ALARM_CLEAR	0x6200	BS2_EVENT_ZONE_APB_VIOLATION
	BS2_EVENT_ZONE_TIMED_APB_VIOLATION	0x6300	TIMED APB
	BS2_EVENT_ZONE_TIMED_APB_ALARM	0x6400	BS2_EVENT_ZONE_TIMED_APB_VIOLATION
	BS2_EVENT_ZONE_TIMED_APB_ALARM_CLEAR	0x6500	BS2_EVENT_ZONE_TIMED_APB_VIOLATION
	BS2_EVENT_ZONE_FIRE_ALARM_INPUT	0x6600	
	BS2_EVENT_ZONE_FIRE_ALARM	0x6700	BS2_EVENT_ZONE_FIRE_ALARM_INPUT
	BS2_EVENT_ZONE_FIRE_ALARM_CLEAR	0x6800	BS2_EVENT_ZONE_FIRE_ALARM_INPUT
	BS2_EVENT_ZONE_FORCED_LOCK_VIOLATION	0x6900	
	BS2_EVENT_ZONE_FORCED_LOCK_START	0x6A00	
	BS2_EVENT_ZONE_FORCED_LOCK_END	0x6B00	
	BS2_EVENT_ZONE_FORCED_UNLOCK_START	0x6C00	
	BS2_EVENT_ZONE_FORCED_UNLOCK_END	0x6D00	
	BS2_EVENT_ZONE_FORCED_LOCK_ALARM	0x6E00	
	BS2_EVENT_ZONE_FORCED_LOCK_ALARM_CLEAR	0x6F00	

8. *param*

,
. .

	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
BioStation 2	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*

.

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   . mask   ID(User, card, door, zone)   .

0	
1	BS2EventExtInfo
2	User ID
4	Card ID
8	Door ID
16	Zone ID
32	BS2EventExtIoDevice
64	Door ID
128	Zone ID
256	TNA Key
512	Job Code
1024	Image
65535	ALL

2. *id*

                              1   .

3. *info*
BS2EventExtInfo   .

4. *userID*

                                                      0   .

5. *cardID*
                card   card   0   .

6. *doorID*
                door   door   0   .

7. *zoneID*
                zone   zone   0   .

8. *ioDevice*
Door   Input   Door   Input   0   .
(BS2EventExtIoDevice   )

9. *tnaKey*

                                                          ,

10. *jobCode*
                JobCode   , JobCode   JobCode
          .

11. *imageSize*

             size   .

12. *image*

13. *reserved*

    .

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*

                , UTC   (sec)   .

2. *deviceID*

            .

3. *subCode*

           .

4. *mainCode*

         .

5. *reserved*

     .

BS2EventExtIoDevice

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

```
} BS2EventExtInfo;
```

1. *ioDeviceID*
Door Input Door Input O .

2. *port*
port number .

3. *value*
port .

-1	UNKNOWN
0	Open
1	Closed
2	Supervised Short
3	Supervised Open

4. *reserved*
.

From:
<https://kb.supremainc.com/bs2sdk/> - **BioStar Device SDK**

Permanent link:
<https://kb.supremainc.com/bs2sdk/doku.php?id=ko:log_management_api&rev=1519354821>

Last update: **2018/02/23 12:00**