

# Table of Contents

- Access Control API** ..... 1
- Structure** ..... 1
- BS2AccessGroup ..... 1
- BS2DoorSchedule ..... 2
- BS2AccessLevel ..... 2
- BS2TimePeriod ..... 2
- BS2DaySchedule ..... 3
- BS2WeeklySchedule ..... 3
- BS2DailySchedule ..... 3
- BS2HolidaySchedule ..... 4
- BS2Schedule ..... 4
- BS2Holiday ..... 5
- BS2HolidayGroup ..... 5

# Access Control API

API for access control, which can manage access authority by individual, group, time schedules.

- [BS2\\_GetAccessGroup](#): Retrieves selected access groups.
- [BS2\\_GetAllAccessGroup](#): Retrieves all access groups.
- [BS2\\_SetAccessGroup](#): Configures an access group.
- [BS2\\_RemoveAccessGroup](#): Removes selected access groups.
- [BS2\\_RemoveAllAccessGroup](#): Removes all access groups.
- [BS2\\_GetAccessLevel](#): Retrieves selected access levels.
- [BS2\\_GetAllAccessLevel](#): Retrieves all access levels.
- [BS2\\_SetAccessLevel](#): Configures an access level.
- [BS2\\_RemoveAccessLevel](#): Removes selected access levels.
- [BS2\\_RemoveAllAccessLevel](#): Removes all access levels.
- [BS2\\_GetAccessSchedule](#): Retrieves selected time schedules.
- [BS2\\_GetAllAccessSchedule](#): Retrieves all time schedules.
- [BS2\\_SetAccessSchedule](#): Configures a time schedule.
- [BS2\\_RemoveAccessSchedule](#): Removes selected time schedules.
- [BS2\\_RemoveAllAccessSchedule](#): Removes all time schedules.
- [BS2\\_GetHolidayGroup](#): Retrieves selected holiday groups.
- [BS2\\_GetAllHolidayGroup](#): Retrieves all holiday groups.
- [BS2\\_SetHolidayGroup](#): Configures a holiday group.
- [BS2\\_RemoveHolidayGroup](#): Removes selected holiday groups.
- [BS2\\_RemoveAllHolidayGroup](#): Removes all holiday groups.

## Structure

### BS2AccessGroup

```
typedef struct {
    uint32_t id;
    char name[BS2_MAX_ACCESS_GROUP_NAME_LEN];
    uint8_t numAccessLevels;
    uint8_t reserved[3];
    uint32_t accessLevels[BS2_MAX_ACCESS_LEVEL_PER_ACCESS_GROUP];
} BS2AccessGroup;
```

1. *id*

Access group ID.

2. *name*

Name of the access group that will be displayed on the BioStar application.

3. *numAccessLevels*

Number of access levels that belongs to an access group.

4. *reserved*

Reserved space.

#### 5. *accessLevels*

List of access levels that belongs to an access group.

## BS2DoorSchedule

```
typedef struct {
    uint32_t doorID;
    uint32_t scheduleID;
} BS2DoorSchedule ;
```

#### 1. *doorID*

Door ID.

#### 2. *scheduleID*

Schedule ID.

## BS2AccessLevel

```
typedef struct {
    uint32_t id;
    char name[BS2_MAX_ACCESS_GROUP_NAME_LEN];
    uint8_t numDoorSchedules;
    uint8_t reserved[3];
    BS2DoorSchedule doorSchedules[BS2_MAX_ACCESS_LEVEL_PER_ACCESS_GROUP];
} BS2AccessLevel;
```

#### 1. *id*

Access level ID. The ID should be a number less than 32768(0x7FFF).

This is due to a limitation where this ID needs to be shared with floor levels for lift control.

#### 2. *name*

Name of the access level that will be displayed on the BioStar application.

#### 3. *numDoorSchedules*

Number of schedules allocated to a door that belongs to an access level.

#### 4. *reserved*

Reserved space.

#### 5. *doorSchedules*

List of schedules allocated to a door that belongs to an access level.

## BS2TimePeriod

```
typedef struct {
```

```
    int16_t startTime;  
    int16_t endTime;  
} BS2TimePeriod;
```

#### 1. *startTime*

Starting time of the time period. The unit of the value is minutes.

#### 2. *endTime*

Ending time of the time period. The unit of the value is minutes.

## BS2DaySchedule

```
typedef struct {  
    uint8_t numPeriods;  
    uint8_t reserved[3];  
    BS2TimePeriod periods[BS2_MAX_TIME_PERIODS_PER_DAY];  
} BS2DaySchedule;
```

#### 1. *numPeriods*

Number of time periods.

#### 2. *reserved*

Reserved space.

#### 3. *periods*

List of time periods, which can be configured up to 5 time periods.

## BS2WeeklySchedule

```
typedef struct {  
    BS2DaySchedule schedule[BS2_NUM_WEEKDAYS];  
} BS2WeeklySchedule;
```

#### 1. *schedule*

Weekly schedule that has total 7 daily schedules.

## BS2DailySchedule

```
typedef struct {  
    uint32_t startDate;  
    uint8_t numDays;  
    uint8_t reserved[3];  
    BS2DaySchedule schedule[BS2_MAX_DAYS_PER_DAILY_SCHEDULE];  
} BS2DailySchedule;
```

#### 1. *startDate*

Starting date of the schedule.

## 2. *numDays*

Number of the daily schedule.

## 3. *reserved*

Reserved space.

## 4. *schedule*

List of daily schedules starting from the *startDate*.

## BS2HolidaySchedule

```
typedef struct {
    uint32_t id;
    BS2DaySchedule schedule;
} BS2HolidaySchedule;
```

### 1. *startDate*

Holiday schedule ID.

### 2. *schedule*

Holiday daily schedule.

## BS2Schedule

```
typedef struct
{
    uint32_t id;
    char name[BS2_MAX_SCHEDULE_NAME_LEN];
    uint8_t isDaily;
    uint8_t numHolidaySchedules;
    uint8_t reserved[2];
    union
    {
        BS2WeeklySchedule weekly;
        BS2DailySchedule daily;
    }schedule;
    BS2HolidaySchedule
    holidaySchedules[BS2_MAX_HOLIDAY_GROUPS_PER_SCHEDULE];
}BS2Schedule;
```

### 1. *id*

Schedule ID.

### CAUTION

You should avoid using 0 and 1 as the schedule ID.

Schedule ID 0 is recognized as 'Not used' and Schedule ID 1 is recognized as 'Always' for every

device.

## 2. *name*

Name of the schedule that will be displayed on the BioStar application.

## 3. *isDaily*

Decides whether it's a daily schedule or a weekly schedule.

## 4. *numHolidaySchedules*

Number of holiday schedules.

## 5. *reserved*

Reserved space.

## 6. *weekly*

Weekly schedule. This will be valid only when *isDaily* is set to 0.

## 7. *daily*

Daily schedule. This will be valid when *isDaily* has a value rather than 0.

## 8. *holidaySchedules*

List of holiday schedules.

## BS2Holiday

```
typedef struct {
    uint32_t date;
    uint8_t recurrence;
} BS2Holiday;
```

### 1. *date*

Date of the holiday.

### 2. *recurrence*

The value of repetition.

Value	Description
0	Does not repeat
1	Repeat yearly
2	Repeat monthly
3	Repeat weekly

## BS2HolidayGroup

```
typedef struct
{
    uint32_t id;
```

```
char name[BS2_MAX_SCHEDULE_NAME_LEN];
uint8_t numHolidays;
uint8_t reserved[3];
BS2Holiday holidays[BS2_MAX_HOLIDAYS_PER_GROUP];
}BS2HolidayGroup;
```

1. *id*

Holiday group ID.

2. *name*

Name of the holiday group that will be displayed on the BioStar application.

3. *numHolidays*

Number of holidays.

4. *reserved*

Reserved space.

5. *holidays*

List of holidays.

From:

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

Permanent link:

[https://kb.supremainc.com/kbtest/doku.php?id=en:access\\_control\\_api](https://kb.supremainc.com/kbtest/doku.php?id=en:access_control_api)

Last update: **2024/10/24 14:45**