Table of Contents

Face API 1
FaceEx API 1
Structure 1
BS2Face 1
BS2AuthGroup 2
BS2TemplateEx 2
BS2FaceEx 3

Face API

This API allows to scan/verify the face template.

- BS2_ScanFace: `FaceStation2` `FaceLite` Scans the face from a device and extracts template and image data.
- BS2_GetAuthGroup: Retrieves selected authentication groups.
- BS2_GetAllAuthGroup: Retrieves all authentication groups.
- BS2_SetAuthGroup: Configures an authentication group.
- BS2_RemoveAuthGroup: Removes selected authentication groups.
- BS2_RemoveAllAuthGroup: Remove all authentication groups.

FaceEx API

- BS2_ScanFaceEx: `FaceStation F2` `BioStation 3` Scans the face from a device and extracts template and image data. [+ 2.7.1]
- BS2_ExtractTemplateFaceEx: `FaceStation F2` `BioStation 3` Extracts template data by the face image. [+ 2.7.1]
- BS2_GetNormalizedImageFaceEx: `FaceStation F2` `BioStation 3` Create a WARP image with an un-warped (unprocessed) face image. [+ 2.8]

Structure

BS2Face

```
typedef struct {
    uint8_t faceIndex;
    uint8_t numOfTemplate;
    uint8_t flag;
    uint8_t reserved;

    uint16_t imageLen;
    uint8_t reserved2[2];

    uint8_t imageData[BS2_FACE_IMAGE_SIZE];
    uint8_t templateData[BS2_TEMPLATE_PER_FACE][BS2_FACE_TEMPLATE_LENGTH];
} BS2Face;
```

1. *faceIndex*
Index of face

2. *numOfTemplate*
Number of face templates.

3. *flag*

This values is used inside the device based on FaceStation2 V1.4.0 and FaceLite V1.2.0.
The flag value received from the device can be used, but if it is not known, it can be set to 0.

4. *reserved*
Reserved space.

5. *imageLen*
Size of the face image.

6. *reserved2*
Reserve space.

4. *imageData*
Face image data.

5. *data*
Face template data.

BS2AuthGroup

```
typedef struct {
    BS2_AUTH_GROUP_ID        id;
    char                name[BS2_MAX_AUTH_GROUP_NAME_LEN];
    uint8_t                reserved[32];
} BS2AuthGroup;
```

1. *id*
Group ID for group matching.

2. *name*
Name of the matching group that will be displayed on BioStar 2

3. *reserved*
Reserved space.

BS2TemplateEx

```
typedef struct {
    uint8_t       data[552];
    uint8_t       isIR;
    uint8_t       reserved[3];
} BS2TemplateEx;
```

`FaceStation F2` `BioStation 3`

1. *data*
IR or visual image template data

2. *isIR*
True when it comes to an IR image, false when it comes to a visual image

3. *reserved*
Reserved

BS2FaceEx

```
typedef struct {
    uint8_t       faceIndex;
    uint8_t       numOfTemplate;
    uint8_t       flag;
    uint8_t       reserved;

    uint32_t      imageLen;
    union {
        struct {
            uint16_t irImageLen;
            uint8_t  unused[6];            ///< 6 bytes (packing)
            uint8_t  imageData[BS2_MAX_WARPED_IMAGE_LENGTH];            ///<
40 * 1024 bytes
            uint8_t  irImageData[BS2_MAX_WARPED_IR_IMAGE_LENGTH];       ///<
30 * 1024 bytes
            BS2TemplateEx templateEx[BS2_MAX_TEMPLATES_PER_FACE_EX];    ///<
20 * 556 bytes
        };

        uint8_t   *rawImageData;

        BS2TemplateEx *onlyTemplateEx;
    };
} BS2FaceEx;
```

FaceStation F2 BioStation 3 BioEntry W3

1. *faceIndex*
Index of face

2. *numOfTemplate*
The number of template including Visual, IR

3. *flag*
Flag whether the image is a WARPed image.
WARP is a kind of generalization that extracts the face among the physical image which contains different body parts.
When flag is set to 1, the device refers to 5 pieces of information defined by the struct in the union.
When flag is set to 0, the device refers to rawImageData in the union.
If the user wants to register a face with a random image that is not WARPed,

please set the flag as BS2_FACE_EX_FLAG_NONE(0), set as image data in the address space of rawImageData, and set the size of image data as imageLen.
When this happens, the device will automatically go through the WARP process with rawImageData and fill in the information in the struct.
Please note that rawImageData and struct are tied in an union.
[+ 2.9.6] An option BS2_FACE_EX_FLAG_TEMPLATE_ONLY(0x20) has been added to allow transmitting only the template excluding the facial image when sending facial data.
This is particularly useful in environments where privacy protection is emphasized.
When using this option, the onlyTemplateEx should be allocated with template data for each numOfTemplate.
Additionally, unableToSaveImageOfVisualFace in BS2FaceConfig must be set to 1 (true).

Value	Description
BS2_FACE_EX_FLAG_NONE	0x00
BS2_FACE_EX_FLAG_WARPED	0x01
BS2_FACE_EX_FLAG_TEMPLATE_ONLY	0x20
BS2_FACE_EX_FLAG_ALL	0xFF

4. *reserved*
Reserved

5. *imageLen*
The size of image data

6. *irImageLen*
The size of IR image data

7. *unused*
Unused space. (for packing)

8. *imageData*
WARPed facial image data. This is made by rawImageData automatically.

Device type	Version	FW version	Image size	Image type	`rawImageData`
FaceStation F2	V1	less than 2.0.0	250 * 250	JPG	JPG
FaceStation F2	V2	2.0.0 or later	112 * 112	PNG	JPG, PNG
BioStation 3	V1	All version	112 * 112	PNG	JPG, PNG

9. *irImageData*
IR image data. IR image might not exist in the WARP process.
It is generated automatically when trying authentication.

10. *templateEx*
Template data of Visual or IR image

11. *rawImageData*
Non-WARPed image data. See `imageData`

12. *onlyTemplateEx*
[+ 2.9.6] This represents contiguous memory information of BS2TemplateEx data, which is template information, and must be allocated as many as numOfTemplate.
This is only used when the flag is BS2_FACE_EX_FLAG_TEMPLATE_ONLY.

From:
https://kb.supremainc.com/bs2sdk/ - **BioStar 2 Device SDK**

Permanent link:
https://kb.supremainc.com/bs2sdk/doku.php?id=en:face_api

Last update: **2024/02/27 13:44**