



<http://www.xprgroup.com>

**Desktop reader SDK**

**Version 1.0**

Copyright © 2017 XPR

## Table of Contents

|                                   |          |
|-----------------------------------|----------|
| <b>About .....</b>                | <b>3</b> |
| <b>Readers .....</b>              | <b>4</b> |
| <b>Modules .....</b>              | <b>5</b> |
| <b>Module DesktopReader .....</b> | <b>6</b> |
| Module ConstantsUSB .....         | 7        |
| Class USBEventArgs .....          | 12       |
| Class Reader .....                | 13       |
| Reader Properties .....           | 13       |
| Reader Functions .....            | 14       |
| Reader Events .....               | 18       |

## About

---

This SDK provide a way to use USB desktop card and fingerprint readers in .NET applications. SDK will enable you to read Card ID number, Read/Write Mifare classic cards, get finger templates, verifying fingerprints using two templates, identifying fingerprints using the template array. Listed possibilities are different for each reader model. Check your requirements before you decide which model to work with. SDK is build with Microsoft Visual Studio 2015 using VB.NET and .NET 4.0. With the SDK is provided VB.NET and C# sample project that will demonstrate full functionality of the SDK.

Support email: [techsupport@xprgroup.com](mailto:techsupport@xprgroup.com)

## Readers

| Reader model                          | PROXE   | PROX-USB  | BIOPROX-USB  | B100PROX-USB  |
|---------------------------------------|---|---|--|---|
|                                       |  |  |  |  |
| <b>Card compatibility</b>             | EM4100<br>Mifare  | EM4100<br>Mifare<br>HID 125KHz<br>AWID  | EM4100<br>Mifare<br>HID 125KHz<br>AWID   | EM4100<br>Mifare<br>HID 125KHz<br>AWID  |
| <b>Mifare Classic 1K/4K Block R/W</b> | Yes   | Yes   | Yes  | Yes   |
| <b>Fingerprint reader</b>             | No  | No  | Yes  | Yes   |
| <b>Finger enrollment type</b>         | -   | -   | Touch  | Swipe   |
| <b>Finger template type</b>           | -   | -   | Suprema  | Authentec Alpha   |
| <b>Compatible fingerprint readers</b> | -   | -   | BIOC3<br>BIOPROX-EH<br>BIOPROX-MF<br>BIOPAD<br>DINBIOC3                            | B100<br>B100PROX-EH<br>B100PROX-MF<br>B100PAD<br>DINB100                            |
| <b>Template verification 1:1</b>      | -   | -   | Yes  | No  |
| <b>Template verification 1:N</b>      | -   | -   | Yes  | No  |
| <b>Write template on card</b>         | Mifare Classic<br>Mifare Desfire  |   |  |   |
| <b>Interface</b>                      | USB   |   |  |   |
| <b>Windows compatibility</b>          | XP, Vista, 7,8,10   |   |  |   |
| <b>Windows driver</b>                 | Not required  |   |  |   |

## Modules

---

Desktop reader SDK consist of the modules listed in the table bellow and sample projects to describe how to use listed modules.

| Module name           | Usage  |
|-----------------------|--|
| DesktopReader.dll     | Main module to interface desktop reader  |
| Suprema.UFMatcher.dll | Modules used by DesktopReader for finger templates matching. Keep all files in the same folder. This modules are required only if template matching is used with BIOPROX-USB reader. |
| UFMatcher.dll         |  |
| UFLicense.dat         |  |

---

## ***Module DesktopReader***

Module components:

| <b>Class</b>  | <b>Description</b>  |
|---------------|---|
| USBEventArgs  | Provide arguments for reader connected/disconnected from USB port |
| Reader        | Main class for reader usage                                       |
| <b>Module</b> | <b>Description</b>  |
| ConstantsUSB  | Enumerated values and structures used in functions                |

## Module ConstantsUSB

|                    |                             |                              |
|--------------------|-----------------------------|------------------------------|
| <b>Name</b>        | Devices                     |                              |
| <b>Type</b>        | Integer                     |                              |
| <b>Description</b> | Enumeration of reader types |                              |
| <b>Members</b>     | <b>Value</b>                | <b>Refer to</b>              |
| NONE               | 255                         | No reader detected           |
| PROXE              | 30                          | Attached PROXE reader        |
| PROXUSB            | 40                          | Attached PROX-USB reader     |
| B100PROXUSB        | 41                          | Attached B100PROX-USB reader |
| BIOPROXUSB         | 42                          | Attached BIOPROX-USB reader  |

|                    |                                     |                        |
|--------------------|-------------------------------------|------------------------|
| <b>Name</b>        | CardTypes                           |                        |
| <b>Type</b>        | Byte                                |                        |
| <b>Description</b> | Enumeration of supported card types |                        |
| <b>Members</b>     | <b>Value</b>                        | <b>Refer to</b>        |
| EM4100             | 1                                   | EM4100 compatible      |
| MIFARE             | 2                                   | Mifare 13.5 MHz        |
| HID                | 4                                   | HID 125 KHz compatible |
| AWID               | 8                                   | AWID compatible        |

|                    |                                  |                 |
|--------------------|----------------------------------|-----------------|
| <b>Name</b>        | MifareTypes                      |                 |
| <b>Type</b>        | Byte                             |                 |
| <b>Description</b> | Enumeration of Mifare card types |                 |
| <b>Members</b>     | <b>Value</b>                     | <b>Refer to</b> |
| Ultralight         | 0                                |                 |
| Classic_1K         | 8                                |                 |
| Mini               | 9                                |                 |
| Plus_2K            | 16                               |                 |
| Plus_4K            | 17                               |                 |
| Classic_4K         | 24                               |                 |
| Desfire            | 32                               |                 |

|                    |  |                 |
|--------------------|--|-----------------|
| <b>Name</b>        | MifareKeys   |                 |
| <b>Type</b>        | Byte   |                 |
| <b>Description</b> | Enumeration of Keys used for Mifare Classic Read/Write operation |                 |
| <b>Members</b>     | <b>Value</b>   | <b>Refer to</b> |
| KeyA               | 96   | Key A           |
| KeyB               | 97   | Key B           |

|                    |                                   |  |
|--------------------|-----------------------------------|--|
| <b>Name</b>        | FingerTemplateTypes               |  |
| <b>Type</b>        | Byte                              |  |
| <b>Description</b> | Finger template types enumeration |  |
| <b>Members</b>     | <b>Value</b>                      | <b>Refer to</b>                            |
| Suprema            | 100                               | Template is taken with BIOPROX-USB reader  |
| DP                 | 201                               | Template is taken with B100PROX-USB reader |

|  |                          |   |
|--|--------------------------|---|
| <b>Name</b>                            | ErrorCodes               |   |
| <b>Type</b>                            | Integer                  |   |
| <b>Description</b>                     | Error returned by reader |   |
| <b>Members</b>                         | <b>Value</b>             | <b>Refer to</b>   |
| NoResponseFromReader                   | 0                        | Reader is not responding. This is the only message not returned from desktop reader |
| ExceptionRaised                        | 100                      |   |
| PutFinger                              | 180                      |   |
| ClassicWriteZerosFail_error_02         | 181                      |   |
| ClassicWriteZerosFail_error_01         | 182                      |   |
| Cannotacces_sector_AccesDenied         | 183                      |   |
| Card_Is_locked                         | 184                      |   |
| Format_authenticate_error              | 185                      |   |
| Format_error_01                        | 186                      |   |
| Card_Already_Unocked_error             | 187                      |   |
| Card_MasterKey_CanNot_Be_Managed_error | 188                      |   |
| Card_Already_Locked_error              | 189                      |   |
| CardFingerLimit_error                  | 190                      |   |
| ClassicWriteBlockFail_error_02         | 191                      |   |
| ClassicWriteBlockFail_error_01         | 192                      |   |
| ClassicNoFingerData                    | 193                      |   |
| ClassicReadBlock_error_02              | 194                      |   |
| ClassicReadBlock_error_01              | 195                      |   |
| ClassicAuthenticate_error_03           | 196                      |   |
| ClassicAuthenticate_error_02           | 197                      |   |
| ClassicAuthenticate_error_01           | 198                      |   |
| ThirdPartyUsesAppNumber_error          | 199                      |   |
| NoApplicationInstalled                 | 200                      |   |
| FormatAuthWSytechKey_error             | 201                      |   |
| FormatAuthWDefKey_error                | 202                      |   |
| Changekey_error_02                     | 203                      |   |
| Changekey_error_01                     | 204                      |   |
| FormatSelectApp_Error_02               | 205                      |   |
| FormatSelectApp_Error_01               | 206                      |   |
| CreateAPP_Error_02                     | 207                      |   |
| CreateAPP_Error_01                     | 208                      |   |
| Authenticate_DES_Error_06              | 209                      |   |

| Name                              | ErrorCodes |                                 |
|-----------------------------------|------------|---------------------------------|
| Authenticate_DES_Error_05         | 210        |                                 |
| Authenticate_DES_Error_04         | 211        |                                 |
| Authenticate_DES_Error_03         | 212        |                                 |
| Authenticate_DES_Error_02         | 213        |                                 |
| Authenticate_DES_Error_01         | 214        |                                 |
| DESCreateFile_Error_02            | 215        |                                 |
| DESCreateFile_Error_01            | 216        |                                 |
| DESListFiles_Error_02             | 217        |                                 |
| DESListFiles_Error_01             | 218        |                                 |
| DESRead_Error_02                  | 219        |                                 |
| DESWrite_Error_04                 | 220        |                                 |
| DESWrite_Error_03                 | 221        |                                 |
| DESWrite_Error_02                 | 222        |                                 |
| DESRead_Error_01                  | 223        |                                 |
| DESWrite_Error_01                 | 224        |                                 |
| DESAuthentication_Error_01        | 225        |                                 |
| DESFormatting_Error_01            | 226        |                                 |
| SelectApp_Error_02                | 227        |                                 |
| SelectApp_Error_01                | 228        |                                 |
| PPS_Error                         | 229        |                                 |
| RATS_Error                        | 230        |                                 |
| UnknownCard                       | 231        |                                 |
| NoCard                            | 232        |                                 |
| Protocol_Error_incomplete_message | 233        |                                 |
| template_chksum_Error             | 239        |                                 |
| receiving_template_fail           | 240        |                                 |
| Scan_fail                         | 241        |                                 |
| Try_Again                         | 242        |                                 |
| Timeout                           | 243        |                                 |
| No_response_from_suprema          | 244        |                                 |
| Bad_firmware_CRC                  | 245        |                                 |
| Bad_firmware_Header               | 246        |                                 |
| Cannot_be_performed               | 249        |                                 |
| Unknown_command                   | 250        |                                 |
| Wrong_values                      | 253        |                                 |
| Checksum_Error                    | 254        |                                 |
| OK                                | 255        | Operation finished with success |

| Name        | MatchErrorCodes   |          |
|-------------|---|----------|
| Type        | Integer   |          |
| Description | Error code returned by templates matching functions<br>CompareTemplates and FindMatch |          |
| Members     | Value   | Refer to |

| Name                   | MatchErrorCodes |                                 |
|------------------------|-----------------|---------------------------------|
| OK                     | 0               | Operation finished with success |
| ERR_GENERAL            | -1              |                                 |
| ERR_NO_LICENSE         | -101            | Missing UFLicense.dat file      |
| ERR_LICENSE_NOT_MATCH  | -102            | Template is not Suprema format  |
| ERR_LICENSE_EXPIRED    | -103            |                                 |
| ERR_NOT_SUPPORTED      | -111            |                                 |
| ERR_INVALID_PARAMETERS | -112            |                                 |
| ERR_MATCH_TIMEOUT      | -401            |                                 |
| ERR_MATCH_ABORTED      | -402            |                                 |
| ERR_TEMPLATE_TYPE      | -411            |                                 |
| ERR_EXCEPTION          | 1               | Exception raised                |

| Name            | ReadIDResault                             |   |
|-----------------|---|---|
| Type            | Structure                                 |   |
| Description     | Structure returned by function ReadCardID |   |
| Members         | Type                                      | Refer to  |
| ErrorCode       | Byte                                      | Success of the operation. Values are enumerated in ErrorCodes   |
| CardType        | Byte                                      | Card type readed.Values are enumerated in CardTypes   |
| UIDLength       | Integer                                   | Length of the ID in bits  |
| MifareType      | Byte                                      | Type of Mifare card reader.Values are enumerated in MifareTypes   |
| ID              | Byte()                                    | Byte array with card ID. Array size is 10   |
| IDConverted7to4 | Byte()                                    | Mifare ID with the length of 7 bytes converted to 4 bytes with NXP conversion for avoiding duplicate card ID between 7 and 4 bytes Mifare cards. Array length is 4. |

| Name        | ReadBlockResault                                      |   |
|-------------|---|---|
| Type        | Structure   |   |
| Description | Structure returned by function ReadMifareClassicBlock |   |
| Members     | Type  | Refer to  |
| ErrorCode   | Byte  | Success of the operation. Values are enumerated in ErrorCodes |
| BlockData   | Byte()  | Array with data from Mifare card. Array length is 16.         |

| Name        | FingerData  |   |
|-------------|---|---|
| Type        | Structure   |   |
| Description | Structure returned by function GetFingerTemplate and ReadFingerFromCard |   |
| Members     | Type  | Refer to  |
| ErrorCode   | Byte  | Success of the operation. Values are enumerated in ErrorCodes |

|              |            |   |
|--------------|------------|---|
| <b>Name</b>  | FingerData |   |
| TemplateType | Byte       | Type of finger template enumerated in FingerTemplateTypes                 |
| Quality      | Byte       | Quality of the Suprema template in percent. Not relevant for DP template. |
| Template     | Byte()     | Finger template. Array size is 384.                                       |

|                    |   |  |
|--------------------|---|--|
| <b>Name</b>        | MatchResault                                    |  |
| <b>Type</b>        | Structure                                       |  |
| <b>Description</b> | Structure returned by function CompareTemplates |  |
| <b>Members</b>     | <b>Type</b>                                     | <b>Refer to</b>  |
| ErrorCode          | Integer   | Success of the operation. Values are enumerated in MatchErrorCodes |
| Match              | Boolean   | Match result   |

|                    |  |  |
|--------------------|--|--|
| <b>Name</b>        | FindMatchResault                         |  |
| <b>Type</b>        | Structure                                |  |
| <b>Description</b> | Structure returned by function FindMatch |  |
| <b>Members</b>     | <b>Type</b>                              | <b>Refer to</b>  |
| ErrorCode          | Integer                                  | Success of the operation. Values are enumerated in MatchErrorCodes |
| MatchIndex         | Integer                                  | Index of matching template. Match found if MatchIndex >= 0.        |

---

## Class USBEventArgs

Class is argument in USBEvent event.

| Parameter | Type    | Description                       |
|-----------|---------|-----------------------------------|
| device    | Integer | Reader type enumerated in Devices |
| version   | String  | Reader firmware version           |

## Class Reader

Reader is the main class in SDK. This class provide all functions for using desktop reader.

### Reader Properties

| Reader properties     |         |                       |   |
|-----------------------|---------|-----------------------|---|
| Property              | Type    | Default value         | Description   |
| AttachedDevice        | Integer | 255<br>(Devices.NONE) | Currently connected device. Possible values are enumerated in Devices |
| AttachedDeviceVersion | String  | "0"                   | Reader firmware version   |
| SecurityLevel         | Integer | 4                     | Matching template FAR (False Accept Ratio)                            |
| FastMode              | Boolean | True                  | Speed mode for function FindMatch                                     |

| Relation between security level and false accept ratio |                          |
|--|--------------------------|
| Level  | FAR (False Accept Ratio) |
| 1  | Below 1 %                |
| 2  | Below 0.1 %              |
| 3  | Below 0.01 %             |
| 4  | Below 0.001 %            |
| 5  | Below 0.0001 %           |
| 6  | Below 0.00001 %          |
| 7  | Below 0.000001 %         |

## Reader Functions

| Function                     | Purpose   |
|------------------------------|---|
| ReadCardID                   | Get card number   |
| ReadMifareClassicBlock       | Read data from one block on Mifare classic card                   |
| WriteMifareClassicBlock      | Write data to one block on Mifare classic card                    |
| GetFingerTemplate            | Scan live finger for template                                     |
| WriteFingerTemplateToCard    | Write finger template to Mifare Classic or Desfire card           |
| ReadFingerFromCard           | Read finger template from Mifare Classic or Desfire card          |
| DeleteFingerData             | Delete finger template from Mifare Classic or Desfire card        |
| UninstallFingerprintFromCard | Remove templates from card and reset passwords to factory default |
| CompareTemplates             | Compare two templates   |
| FindMatch                    | Find match of template in templates array                         |

### ○ Function ReadCardID

#### Declaration:

```
Public Function ReadCardID(CardType AsByte) As ReadIDResault
```

#### Parameters:

- CardType:  
Type of the card reader should expect. Possible values are listed in Constants.CardTypes.

#### Return value:

Function returns structure ReadIDResault. If the value of the structure member ReadIDResault.ErrorCode is 255 (ErrorCodes.OK) then structure contains valid card data.

When reading HID cards, regardless of the card ID length, function will return all bits read from the card except the last significant bit.

### ○ Function ReadMifareClassicBlock

#### Declaration:

```
PublicFunction ReadMifareClassicBlock(KeyType AsByte, key AsByte(), Sector AsByte, block AsByte) AsReadBlockResault
```

#### Parameters:

- .KeyType:  
Possible values are enumerated in Constants.MifareKeys. This parameter defines the sector key to be used to get the block data.
- Key:  
Password for sector access. Array is with 6 elements.
- Sector:  
Sector of the card
- block:  
Block in the sector to read

#### Return value:

Function returns structure ReadBlockResault. If the value of the structure member

ReadBlockResult.ErrorCode is 255 (ErrorCodes.OK) then structure contains valid block data as byte array of 16 elements.

#### ○ Function WriteMifareClassicBlock

##### Declaration:

**PublicFunction** WriteMifareClassicBlock(KeyType **AsByte**, key **AsByte**(), Sector **AsByte**, block **AsByte**, Data **AsByte**()) **AsByte**

##### Parameters:

- .KeyType:  
Possible values are enumerated in Constants.MifareKeys. This parameter defines the sector key to be used to write data in the block.
- Key:  
Password for sector access. Array is with 6 elements.
- Sector:  
Sector of the card
- block:  
Block in the sector to write
- Data  
Byte array with 16 elements to be written in the block.

##### Return value:

Possible return values are listed in Constants.ErrorCodes.

##### Important!

**Last block in the sector contains passwords and sector access configuration bits. DO NOT write in the last block of the sectors unless you are sure that new data will not result with the lost of the sector. Last block in the sector can be block 3 or 15 (depends of the card type and sector number) so avoiding this block will keep you on the "safe side" when using this function.**

#### ○ Function GetFingerTemplate

##### Declaration:

**PublicFunction** GetFingerTemplate() **AsFingerData**

##### Return value:

If returned structure member FingerData.ErrorCode is 255 (ErrorCodes.OK) then structure contains valid finger template in FingerData.Template as byte array with 384 elements.

#### ○ Function WriteFingerTemplateToCard

##### Declaration:

**PublicFunction** WriteFingerTemplateToCard(FingerNumber **AsByte**, Template **AsByte**()) **AsByte**

##### Parameters:

- .FingerNumber:  
Depending of the card memory, cards can have one or two fingers. For Mifare Classic 1K FingerNumber must be 1 while for Mifare Classic 4K and Desfire FingerNumber can be 1 or 2.
- Template as byte array of 384 elements.

##### Return value:

Possible return values are listed in Constants.ErrorCodes. Value of 255 (ErrorCodes.OK) means successful operation.

### ○ Function ReadFingerFromCard

#### Declaration:

`PublicFunction ReadFingerFromCard(FingerNumber AsByte) AsFingerData`

#### Parameters:

- .FingerNumber

Depending of the card memory, cards can have one or two fingers. For Mifare Classic 1K FingerNumber must be 1 while for Mifare Classic 4K and Desfire FingerNumber can be 1 or 2.

#### Return value:

If returned structure member FingerData.ErrorCode is 255 (ErrorCodes.OK) then structure contains valid finger template in FingerData.Template as byte array with 384 elements.

### ○ Function DeleteFingerData

#### Declaration:

`PublicFunction DeleteFingerData(FingerNumber AsByte) AsByte`

#### Parameters:

- .FingerNumber

Depending of the card memory, cards can have one or two fingers. For Mifare Classic 1K FingerNumber must be 1 while for Mifare Classic 4K and Desfire FingerNumber can be 1 or 2.

#### Return value:

Possible return values are listed in Constants.ErrorCodes. Value of 255 (ErrorCodes.OK) means successful operation.

### ○ Function UninstallFingerprintFromCard

#### Declaration:

`PublicFunction UninstallFingerprintFromCard() AsByte`

#### Return value:

Possible return values are listed in Constants.ErrorCodes. Value of 255 (ErrorCodes.OK) means successful operation.

### ○ Function CompareTemplates

#### Declaration:

`PublicFunction CompareTemplates(Template1 AsByte(), Template2 AsByte(), SecurityLevel AsInteger) AsMatchResault`

#### Parameters:

- .Template1, Template2:  
Templates to be matched
- SecurityLevel

Values are explained in chapter "Reader Properties", property SecurityLevel.

#### Return value:

If returned structure member MatchResault.ErrorCode is 0 (MatchErrorCodes.OK) then MatchResault.Match will hold the information if templates are matching.

### ○ **Function FindMatch**

**Declaration:**

`PublicFunction FindMatch(Template1 AsByte(), Template2Array AsByte()), Timeout AsInteger)  
AsFindMatchResault`

**Parameters:**

- .Template1:  
Template to be matched with templates array Template2Array
- Template2Array:  
Array of templates to be searched for matching
- Timeout:  
Timeout for operation in milliseconds.

**Return value:**

If returned structure member FindMatchResault.ErrorCode is 0 (MatchErrorCodes.OK) then FindMatchResault.MatchIndex will hold the index of the matching template in the Template2Array.

---

## Reader Events

| Event    | Rise when                                 |
|----------|---|
| USBEvent | Desktop reader is plugged/removed from PC |

### ○ Event USBEvent

Declaration:

```
PublicEvent USBEvent(sender AsObject, e AsUSBEventArgs)
```

Parameters:

- sender: DesktopReader.Reader
- e: argument packed in the class USBEventArgs