



# Software Integration Guide

## Topaz SigIDExtLite SDK

Designed for Use in Chrome and Firefox  
Browser Extension Frameworks

**Version 1.1.4.0**

Last Update: January 18, 2019

Copyright © 2019 Topaz Systems Inc. All rights reserved.

For Topaz Systems, Inc. trademarks and patents, visit [www.topazsystems.com/legal](http://www.topazsystems.com/legal).

## Table of Contents

- 1.0 – Introduction..... 4**
  
- 2.0 – Overview and Architecture ..... 4**
  - 2.1 – Topaz SigIDExtLite SDK ..... 5
  - 2.2 – Chrome Extension/Webpage..... 5
  - 2.3 – Firefox Extension/Webpage ..... 5
  
- 3.0 – Key Features ..... 6**
  
- 4.0 – Operating Systems Supported ..... 6**
  
- 5.0 – Fingerprint Capture Devices..... 6**
  
- 6.0 – Instructions to Run Sample Applications..... 7**
  - 6.1 – Chrome..... 7
    - 6.1.1 – Install SigIDp1 .....7*
    - 6.1.2 – Install SigIDExtLite .....7*
    - 6.1.3 – Install the Topaz SigIDExtLite Chrome Extension .....8*
    - 6.1.4 – Run the Sample Web Page .....8*
  
  - 6.2 – Firefox ..... 9
    - 6.2.1 – Install SigIDp1 .....9*
    - 6.2.2 – Install SigIDExtLite .....9*
    - 6.2.3 – Run the Sample Web Page ..... 10*

## Table of Contents

- 7.0 – SigIDExtLite Integration for Fingerprint Capture in Chrome and Firefox ..... 11**
  - 7.1 – Launching the Extensions from a Webpage ..... 11
  
- 8.0 – Fingerprint Capture and Data Export..... 12**
  - 8.1 – Enroll Fingerprint ..... 12
    - 8.1.1 – INPUT Message..... 13**
    - 8.1.2 – OUTPUT Message..... 14**
  
  - 8.2 – Validate Fingerprint ..... 14
    - 8.2.1 – INPUT Message..... 15**
    - 8.2.2 – OUTPUT Message..... 16**
  
  - 8.3 – Capture Fingerprint Image..... 17
    - 8.3.1 – INPUT Message..... 17**
    - 8.3.2 – OUTPUT Message..... 18**
  
  - 8.4 – Capture Fingerprint Image (Low Resolution) ..... 18
    - 8.4.1 – INPUT Message..... 19**
    - 8.4.2 – OUTPUT Message..... 20**
  
- 9.0 – End User Deployment ..... 20**

## 1.0 – Introduction

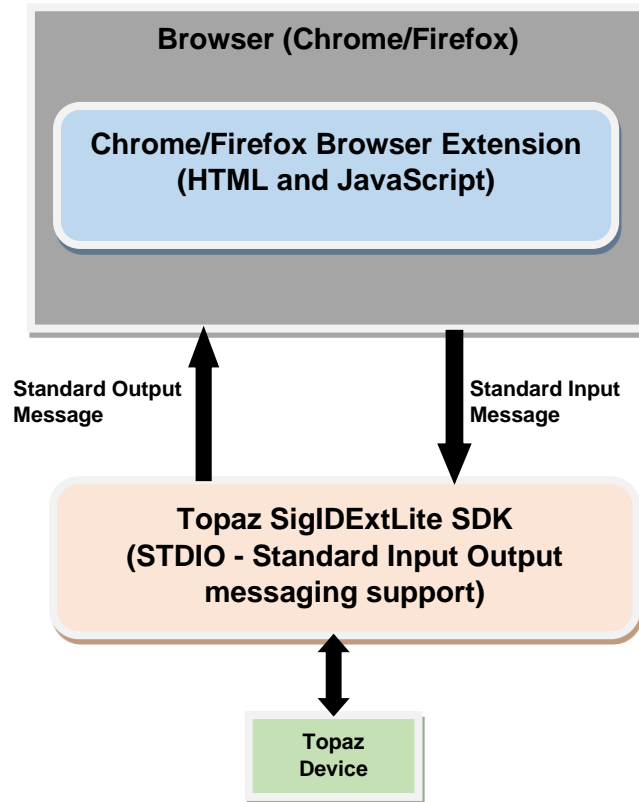
The Topaz SigIDExtLite SDK offers a mechanism and platform for developers and integrators to capture fingerprints securely for web applications running in the Chrome and Firefox browsers. The SDK provides capabilities for enrollment and validation of fingerprints using fingerprint Sensor enabled pads from Topaz Systems.

The SDK exports the images of the captured fingerprints in PNG format. The images can be used in any application requiring fingerprint images.

## 2.0 – Overview and Architecture

As Google Chrome and Mozilla Firefox have discontinued support for plug-ins running inside the browsers, neither Java applets nor NPAPI plug-ins can be used with these browsers. The SigIDExtLite browser extensions are designed to provide web pages with the capability to capture fingerprints using Topaz pads connected to client desktops.

The diagram below shows the high-level overview of the solution with critical components involved.



## 2.1 – Topaz SigIDExtLite SDK

The Topaz SigIDExtLite SDK has been developed as a standard C#.Net application. It has built-in mechanisms to capture fingerprints using Topaz devices for biometric enrollment and verification of users and also to expose the fingerprint as an image (PNG). Its interfaces are implemented as Standard Input and Output streams under the Chrome and Firefox Browser Extension frameworks. The SDK processes the input text messages from the Chrome and Firefox browsers and executes the requests asynchronously, and when a task is complete sends back the status or output data as an output text message. It will host all the User Interface functions for capture and display of fingerprints from devices.

Browsers run this SDK in a separate process, launch it through Google Connect APIs, and send a notification back to the Extension when the application is ended by the user.

## 2.2 – Chrome Extension/Webpage

Chrome Extensions are the HTML, JavaScript, and CSS based code modules that are launched during startup of the browser or launched on demand from web page JavaScript. The Chrome Extensions use JavaScript based Google Native Messaging APIs to launch and communicate with the Topaz SigIDExtLite SDK for fingerprint capture and other relevant features. The extension listens for the output messages from SigIDExtLite and processes them accordingly. Google Native Messaging has a Connect API to launch the applications (which can process the standard input and output messages) in this SigIDExtLite SDK and a Disconnect event to let the web page know about termination of the native host application. Using Connect and Disconnect, the life cycle of the native host application can be controlled. Also, Google Native Messaging APIs have mechanisms to send input messages to the SigIDExtLite SDK and receive output messages from applications.

## 2.3 – Firefox Extension/Webpage

Firefox Extensions are the HTML, JavaScript, and CSS based code modules that are launched during startup of the browser. The Firefox extension uses Chrome Native Messaging APIs to launch the Topaz SigIDExtLite SDK and send and receive input and output messages. It also has the required callback mechanism to notify the calling module when the launched application is terminated or when some error has occurred.

## 3.0 – Key Features

The Topaz SigIDExtLite SDK provides the following features:

- Enroll a fingerprint template
- Capture and validate fingerprints against a specified fingerprint template (captured during “Enroll”)
- Capture and export fingerprint BMP as a high-resolution image in PNG format
- Capture and export fingerprint BMP as a low-resolution image in PNG format
- Specify the position of the fingerprint capture dialog on the computer screen
- Specify allowed number of retries during fingerprint validation against a template

## 4.0 – Operating Systems Supported

The Topaz SigIDExtLite SDK can be integrated into web pages running in the latest versions of the Chrome and Firefox browsers installed on Windows 7 and newer 32-bit and 64-bit operating systems. For 64-bit Windows operating systems, it is recommended to use the 32-bit Chrome and Firefox browser versions.

Note: Microsoft Visual Studio 2010 with .NET Framework 4.0 is the development IDE used in developing the SigIDExtLite SDK, and hence the .NET Framework 4.0 run time should be available in the end user Window’s computer.

## 5.0 – Fingerprint Capture Devices

The SigIDExtLite SDK supports capturing fingerprints using the following devices from Topaz Systems:

- TF-LBK463-HSB-R
- TF-LBK464-HSB-R
- TF-S463-HSB-R

## 6.0 – Instructions to Run Sample Applications

### 6.1 – Chrome

Using SigIDExtLite in Chrome requires three simple steps, as described below:

#### 6.1.1 – Install SigIDp1

Note: Be sure to install SigIDp1 directly from the link below.

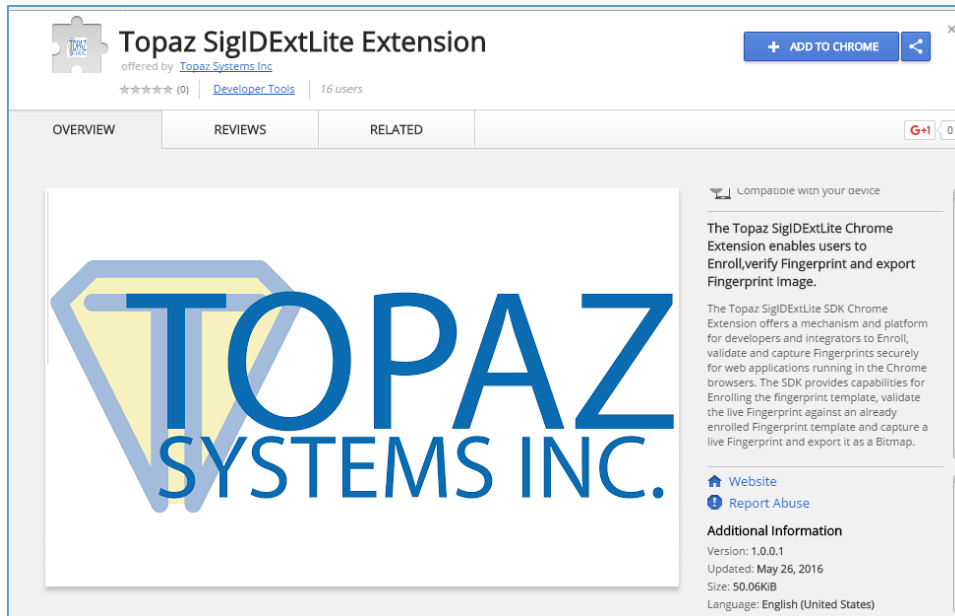
1. Download and install the SigIDp1 installer from the following link: [www.topazsystems.com/Software/sigidp1wp.exe](http://www.topazsystems.com/Software/sigidp1wp.exe). Right-click on the “SigIDp1WP.exe” installer, and choose “Run as Administrator”. Then, follow the steps in the installer.

#### 6.1.2 – Install SigIDExtLite

1. Download and save SigIDExtLite to your computer’s desktop from the following link: [www.topazsystems.com/Software/SigIDExtLite.exe](http://www.topazsystems.com/Software/SigIDExtLite.exe).
2. Right-click on the “SigIDExtLite.exe” installer, and choose “Run as Administrator”. Then, follow the steps in the installer.

### 6.1.3 – Install the Topaz SigIDExtLite Chrome Extension

1. Start the Chrome Browser, and go to the Topaz “SigIDExtLite” Extension page at: <https://chrome.google.com/webstore/detail/topaz-sigidextlite-extens/jdkekdbbdcjhhkddbojnlohjgejcebf>.
2. In the opened page, click on the “+ ADD TO CHROME” button displayed on the top right of the page (see the screenshot below).



3. Click on the “Add extension” button in the popped-up confirmation dialog.

### 6.1.4 – Run the Sample Web Page

After installing the three required SigIDExtLite software and the Topaz signature pad is connected to the client desktop, launch Chrome and navigate to the following page: [www.topazsystems.com/sdks/sigidextlite.html](http://www.topazsystems.com/sdks/sigidextlite.html).

Click on the:

- “Enroll” button to enroll a fingerprint template
- “Validate” button to validate the captured fingerprint against an enrolled template
- “Bitmap” button to capture a fingerprint image in PNG format
- “LowRes Bitmap” to capture a low-resolution image in PNG format

Each of the above buttons will display a fingerprint capture dialog, and the dialog position can be controlled using the FP Window Position, X and Y user input fields. “Validate Retry Limit” sets the number of retries to be allowed during fingerprint validation.



## 6.2 – Firefox

Using SigIDExtLite in Firefox requires two simple steps, as described below:

### 6.2.1 – Install SigIDp1

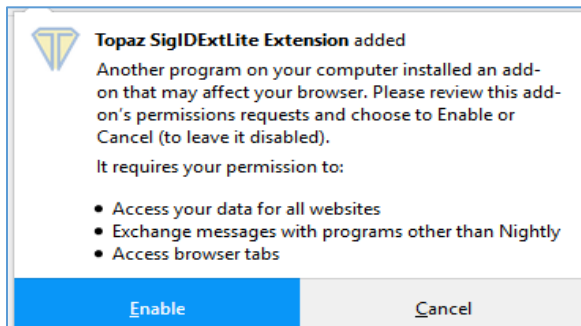
Note: Be sure to install SigIDp1 directly from the link below.

1. Download and install the SigIDp1 installer from the following link: [www.topazsystems.com/Software/sigidp1wp.exe](http://www.topazsystems.com/Software/sigidp1wp.exe). Right-click on the “SigIDp1WP.exe” installer, and choose “Run as Administrator”. Then, follow the steps in the installer.

### 6.2.2 – Install SigIDExtLite

Note: If applicable, remove any previous version of the SigIDExtLite Firefox extension from the “Firefox Add-ons Manager” page under the “Extension” section.

1. Download and save SigIDExtLite to your computer’s desktop from the following link: [www.topazsystems.com/Software/SigIDExtLite.exe](http://www.topazsystems.com/Software/SigIDExtLite.exe).
2. Right-click on the “SigIDExtLite.exe” installer, and choose “Run as Administrator”. Then, follow the steps in the installer.
3. Start the Firefox browser to complete the installation of the Firefox Extension. The browser prompts the user to allow the extension installation. Click on the “Enable” button to complete the installation.



**Note:**

- To see the version of the SDK extension, click on “More”.
- Step 2 has to be performed for all the Windows login accounts by logging into Windows and starting the Firefox browser.
- In some cases, the Firefox browser may not show the prompt mentioned in step 2; in those cases, the extension has to be manually enabled from the Firefox Add-ons manager page by typing in about: addons in the address bar, then going to the Extensions section and clicking on the ‘Enable’ button next to ‘Topaz SigIDExtLite Extension’ and restarting the browser.

**6.2.3 – Run the Sample Web Page**

After installing the three required SigIDExtLite software and the Topaz signature pad is connected to the client desktop, launch Firefox and navigate to the following page: [www.topazsystems.com/sdks/sigidextlite.html](http://www.topazsystems.com/sdks/sigidextlite.html).

Click on the:

- “Enroll” button to enroll a fingerprint template
- “Validate” button to validate the captured fingerprint against an enrolled template
- “Bitmap” button to capture a fingerprint image in PNG format
- “LowRes Bitmap” to capture a low-resolution image in PNG format

Each of the above buttons will display a fingerprint capture dialog, and the dialog position can be controlled using the FP Window Position, X and Y user input fields. “Validate Retry Limit” sets the number of retries to be allowed during fingerprint validation.

## 7.0 – SigIDExtLite Integration for Fingerprint Capture in Chrome and Firefox

For web pages running in Chrome and Firefox the only required step is to raise and listen for predefined custom HTML events within the web page.

### 7.1 – Launching the Extensions from a Webpage

The SigIDExtLite extension relies on custom HTML events for communication between the web page and the extensions and vice versa.

The Chrome and Firefox Extensions loads during browser start up and registers a custom HTML event named “SigIDExtLiteRequestEvent”. Web pages wishing to capture the Fingerprint using the SigIDExtLite SDK in Chrome and Firefox browsers have to raise the custom HTML event “SigIDExtLiteRequestEvent” and send an input message to the SDK as an event attribute.

Once the requested task is completed, the SigIDExtLite extensions raise a custom HTML event named “SigIDExtLiteResponseEvent” and pass the output message as an event attribute. Web pages should register and implement the “SigIDExtLiteResponseEvent” event for processing the output from the extension.

The following code snippet demonstrates raising the custom HTML event “SigIDExtLiteRequestEvent” to initiate Fingerprint capture and also register and implement the “SigIDExtLiteResponseEvent” event for processing the output from the SDK.

```
message = { "requestType": 3, "winPositionMode": winPos, "xPos": xVal, "yPos": yVal };
var messageData = JSON.stringify(message);
document.addEventListener('SigIDExtLiteResponseEvent', SigIDExtLiteResponse, false);
var element = document.createElement("SigIDExtLiteDataElement");
element.setAttribute("SigIDExtLiteRequestAttributes", messageData);
document.documentElement.appendChild(element);
var evt = document.createEvent("Events");
evt.initEvent("SigIDExtLiteRequestEvent", true, false);
element.dispatchEvent(evt);

function SigIDExtLiteResponse(event) {
    var str = event.target.getAttribute("SigIDExtLiteResponseAttributes");
    var obj = JSON.parse(str);
    //Process the response
}
```

## 8.0 – Fingerprint Capture and Data Export

As the SigIDExtLite SDK is designed to support Standard Input and Output streams for communication between the browser extensions and the SDK, only text data can be exchanged between the applications. The Input message triggers the Fingerprint capture module, and the input message itself contains all the required data and type of request as payload.

The Output message payload will contain the request status, the output data enrollment template, Fingerprint image data, and a parameter to carry the error message in case signing fails.

The Chrome Google native messaging API (through which the Extension sends and receives data to the SDK) mandates the text data to be in JSON format, hence the input and output messages should be in JSON format and for convenience the same JSON format is followed for Firefox as well.

The format of the JSON message is

```
{text: value1, text1: value2}
```

where 'text' and 'text1' are the names of the JSON parameters.

### 8.1 – Enroll Fingerprint

Captures multiple Fingerprints from a user and if successful returns the Fingerprint template data as a base64 string and further it can be used for user validation.

### 8.1.1 – INPUT Message

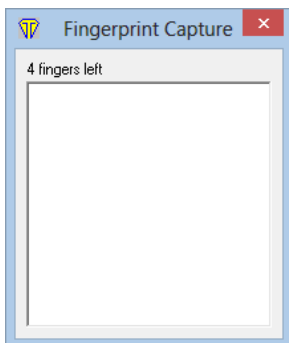
Here is a sample JSON string containing an INPUT message for Enrollment of a Fingerprint template.

```
{ "requestType": 1, "winPositionMode": 1, "xPos": 100, "yPos": 150 }
```

Parameter	Description
<b>requestType</b>	Requested operation as an Integer. The value should be 1 for Fingerprint Enrollment.
<b>winPositionMode</b>	Defines the position of the Fingerprint capture dialog with respect to computer screen. 0 - Default (Auto position) 1 - Manual (set x,y value for window position in px) 2 - CENTER 3 - TOP-LEFT 4 - TOP-CENTER 5 - TOP-RIGHT 6 - RIGHT-CENTER 7 - BOTTOM-RIGHT 8 - BOTTOM-CENTER 9 - BOTTOM-LEFT 10 - LEFT-CENTER
<b>xPos</b>	Position of the Fingerprint capture dialog on X axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1
<b>yPos</b>	Position of the Fingerprint capture dialog on Y axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1

Once the required JSON input message is formatted, you should pass it to the Chrome or Firefox extensions. It should be set as an attribute to the custom html event **SigIDExtLiteRequestEvent** raised by a Web page.

The following interface appears for Fingerprint enrollment. The user has to provide 4 fingerprint samples for successful enrollment.



### 8.1.2 – OUTPUT Message

The SigIDExtLite SDK sends back an output message in the following scenarios

1. Failed to open a connection with the fingerprint device.
2. User cancelled fingerprint enrollment.
3. Fingerprint enrollment successful.
4. Fingerprint enrollment failed.

The output message has a 'status' Boolean parameter indicating whether the Fingerprint Enrollment is successful or not. Applications can rely on this parameter to identify if an Enrollment is successful or not.

Here is a sample output message

```
{"status":false, "message":"Fingerprint device could not be initialized", "outputString":""}
```

Parameter	Description
<b>Status</b>	Status of Fingerprint Enrollment request as Boolean. 'true' indicates successful enrollment and 'false' indicates failed enrollment. If successful the outputString attribute of output JSON message contains the Fingerprint enrollment data as a base 64 string. If failed the message attribute of output JSON message has the reason for failure.
<b>Message</b>	Contains the reason for failure if the request is failed/cancelled, otherwise it will be empty for successful enrollment.
<b>outputString</b>	Carries the Fingerprint Enrollment template data as a Base64 string for successful enrollment and will be empty for failed requests.

The OUTPUT message is sent back as an event attribute for the event **SigIDExtLiteResponseEvent** raised by the extensions and handled by the web page.

### 8.2 – Validate Fingerprint

Captures a Fingerprint from the user and validates it against set Fingerprint template data.

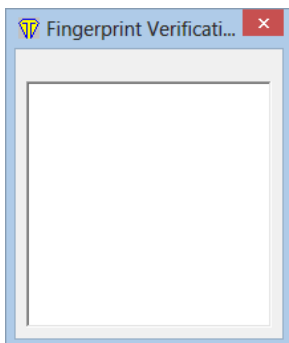
### 8.2.1 – INPUT Message

Here is a sample JSON string containing INPUT message for Validation of a Fingerprint.

```
{ "requestType": 2, "enrollTemplateData": "Fingerprint Enrollment data as base64 String", "winPositionMode": 1, "xPos": 100, "yPos": 200, "validateRetryLimit": 5 };
```

Parameter	Description
<b>requestType</b>	Requested operation as an Integer. The value should be 2 for Fingerprint validation.
<b>enrollTemplateData</b>	Fingerprint template string as returned by Enrollment request.
<b>winPositionMode</b>	Defines the position of the Fingerprint capture dialog with respect to computer screen. 0 - Default (Auto position) 1 - Manual (set x,y value for window position in px) 2 - CENTER 3 - TOP-LEFT 4 - TOP-CENTER 5 - TOP-RIGHT 6 - RIGHT-CENTER 7 - BOTTOM-RIGHT 8 - BOTTOM-CENTER 9 - BOTTOM-LEFT 10 - LEFT-CENTER
<b>xPos</b>	Position of the Fingerprint capture dialog on X axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1
<b>yPos</b>	Position of the Fingerprint capture dialog on Y axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1
<b>validateRetryLimit</b>	Number of failed retries allowed before ending the validation request as Integer. The valid range is between 1 and 10. Any value specified out of this range defaults to value of 5.

Once the required JSON input message is formatted, you should pass it to the Chrome or Firefox extensions. It should be set as an attribute to the custom html event **SigIDExtLiteRequestEvent** raised by a Web page. The following interface appears for Fingerprint validation. The user has to provide a Fingerprint sample for validation against a set template.



### 8.2.2 – OUTPUT Message

The SigIDExtLite SDK sends back an output message in the following scenarios

1. Failed to open a connection with the fingerprint device.
2. User cancelled fingerprint validation.
3. Fingerprint validation successful.
4. Fingerprint validation failed.

The output message has a 'status' Boolean parameter indicating whether the Fingerprint validation is successful or not. Applications can rely on this parameter to identify if an Enrollment is successful or not.

Here is a sample output message

```
{"status":true, "message":""}
```

Parameter	Description
<b>Status</b>	Status of Fingerprint validation request as Boolean. 'true' indicates successful validation, and 'false' indicates failed validation. If validation fails and the status returned is false then the message attribute of output JSON message has the reason for failure
<b>Message</b>	Contains the reason for failure if the request is failed/cancelled otherwise it will be empty for successful enrollment.

The OUTPUT message is sent back as an event attribute for the event **SigIDExtLiteResponseEvent** raised by the extensions and handled by the web page.



### 8.3 – Capture Fingerprint Image

Captures a Fingerprint from the user and exports the Fingerprint bitmap in PNG format.

#### 8.3.1 – INPUT Message

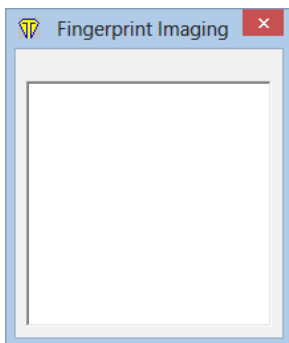
Here is a sample JSON string containing an INPUT message for Fingerprint image capture.

```
{ "requestType": 3, "winPositionMode": 3, "xPos": 0, "yPos": 0 }
```

Parameter	Description
<b>requestType</b>	Requested operation as an Integer. The value should be 3 for Capturing a Fingerprint image.
<b>winPositionMode</b>	Defines the position of the Fingerprint capture dialog with respect to computer screen. 0 - Default (Auto position) 1 - Manual (set x,y value for window position in px) 2 - CENTER 3 - TOP-LEFT 4 - TOP-CENTER 5 - TOP-RIGHT 6 - RIGHT-CENTER 7 - BOTTOM-RIGHT 8 - BOTTOM-CENTER 9 - BOTTOM-LEFT 10 - LEFT-CENTER
<b>xPos</b>	Position of the Fingerprint capture dialog on X axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1
<b>yPos</b>	Position of the Fingerprint capture dialog on Y axis in Pixels. This should be set when the winPositionMode value is set to manual, i.e., 1

Once the required JSON input message is formatted, you should pass it to the Chrome or Firefox extensions. It should be set as an attribute to the custom html event **SigIDExtLiteRequestEvent** raised by a Web page.

The following interface appears for Fingerprint image capture. User has to provide a Fingerprint sample for exporting the image.



### 8.3.2 – OUTPUT Message

The SigIDExtLite SDK sends back an output message in the following scenarios

1. Failed to open a connection with the fingerprint device.
2. User cancelled fingerprint capture.
3. Fingerprint capture successful.
4. Fingerprint capture failed.

The output message has a 'status' Boolean parameter indicating whether the Fingerprint image capture is successful or not. Applications can rely on this parameter to identify if an Enrollment is successful or not.

Here is a sample output message

```
{"status":false, "message":"Fingerprint device could not be initialized", "outputString":""}
```

Parameter	Description
<b>Status</b>	Status of Fingerprint image capture request as Boolean. 'true' indicates successful capture, and 'false' indicates failed or cancelled capture request. If successful the outputString attribute of output JSON message contains the Fingerprint image data as a base 64 string. If failed the message attribute of output JSON message has the reason for failure.
<b>Message</b>	Contains the reason for failure if the request is failed/cancelled, otherwise it will be empty for successful enrollment.
<b>outputString</b>	Carries the Fingerprint image PNG data as a Base64 string for successful capture and will be empty for failed capture.

The OUTPUT message is sent back as an event attribute for the event **SigIDExtLiteResponseEvent** raised by the extensions and handled by the web page.

### 8.4 – Capture Fingerprint Image (Low Resolution)

Captures a Fingerprint from the user and exports the fingerprint low-resolution bitmap in PNG format.

### 8.4.1 – INPUT Message

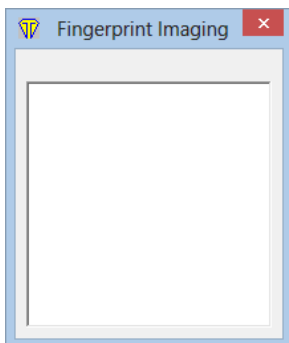
Here is a sample JSON string containing an INPUT message for capturing a low-resolution fingerprint image.

```
{ "requestType": 4, "winPositionMode": 9, "xPos": 0, "yPos": 0 }
```

Parameter	Description
<b>requestType</b>	Requested operation as an Integer. The value should be 4 for Capturing a Fingerprint image in low resolution PNG format
<b>winPositionMode</b>	Defines the position of the Fingerprint capture dialog with respect to computer screen. 0 - Default (Auto position) 1 - Manual (set x,y value for window position in px) 2 - CENTER 3 - TOP-LEFT 4 - TOP-CENTER 5 - TOP-RIGHT 6 - RIGHT-CENTER 7 - BOTTOM-RIGHT 8 - BOTTOM-CENTER 9 - BOTTOM-LEFT 10 - LEFT-CENTER
<b>xPos</b>	Position of the Fingerprint capture dialog on X axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1
<b>yPos</b>	Position of the Fingerprint capture dialog on Y axis in pixels. This should be set when the winPositionMode value is set to manual, i.e., 1

Once the required JSON input message is formatted, you should pass it to the Chrome or Firefox extensions. It should be set as an attribute to the custom html event **SigIDExtLiteRequestEvent** raised by a Web page.

The following interface appears for low-resolution fingerprint image capture. User has to provide a fingerprint sample for image export.



### 8.4.2 – OUTPUT Message

The SigIDExtLite SDK sends back an output message in the following scenarios

1. Failed to open a connection with the fingerprint device.
2. User cancelled fingerprint capture.
3. Fingerprint low resolution image capture successful.
4. Fingerprint low resolution image capture failed.

The output message has a ‘status’ Boolean parameter indicating whether the Fingerprint image capture is successful or not. Applications can rely on this parameter to identify if an Enrollment is successful or not.

Here is a sample output message

```
{"status":false, "message":"Fingerprint device could not be initialized", "outputString":""}
```

Parameter	Description
<b>Status</b>	Status of Fingerprint low-resolution image capture request as Boolean. ‘true’ indicates successful capture, and ‘false’ indicates failed or cancelled capture request. If successful the outputString attribute of output JSON message contains the low-resolution Fingerprint image data as a base 64 string. If failed, the message attribute of output JSON message has the reason for failure.
<b>Message</b>	Contains the reason for failure if the request is failed/cancelled otherwise it will be empty for successful enrollment.
<b>outputString</b>	Carries the low-resolution Fingerprint image PNG data as a Base64 string for successful capture and will be empty for failed capture.

The OUTPUT message is sent back as an event attribute for the event **SigIDExtLiteResponseEvent** raised by the extensions and handled by the web page.

## 9.0 – End User Deployment

Once the Topaz SigIDExtLite SDK integration is completed, the next step is to deploy the required software on end user machines. Follow the steps outlined in the SigIDExtLite User Installation Guide at: [www.topazsystems.com/Software/SigIDExtLite\\_UserInstall.pdf](http://www.topazsystems.com/Software/SigIDExtLite_UserInstall.pdf).