

Table of Contents

How to enroll fingerprint on BioMini via BioStar 2 API	1
Step 1:	1
Step 2:	2
Step 3:	2
Step 4:	2

How to enroll fingerprint on BioMini via BioStar 2 API

You can enroll fingerprint on BioMini by calling BioStar 2 API.

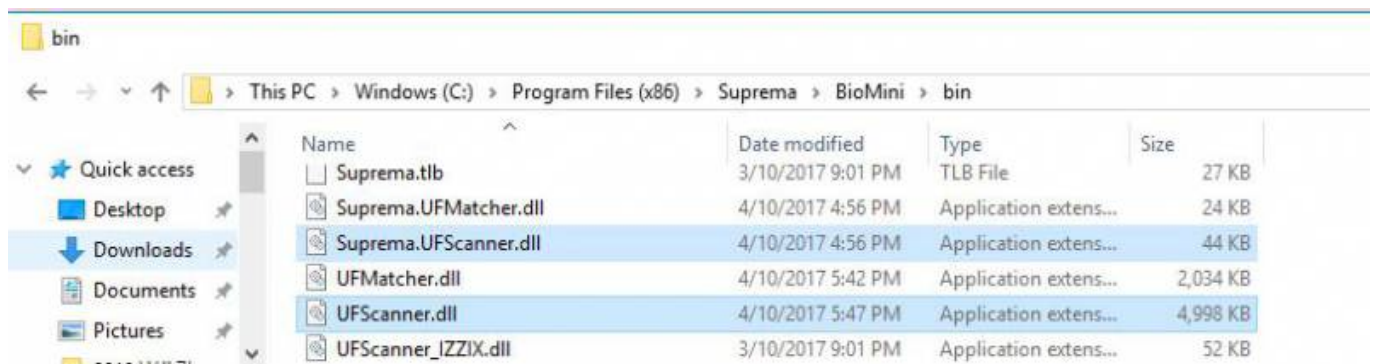
BioMini is only enrollment device but can't authenticate fingerprint. Thus, it doesn't appear in GET/devices.

To utilize BioMini with BioStar 2 API, you should download BioMini SDK and install it. If you need BioMini SDK, contact our sales team (sales_id@supremainc.com).

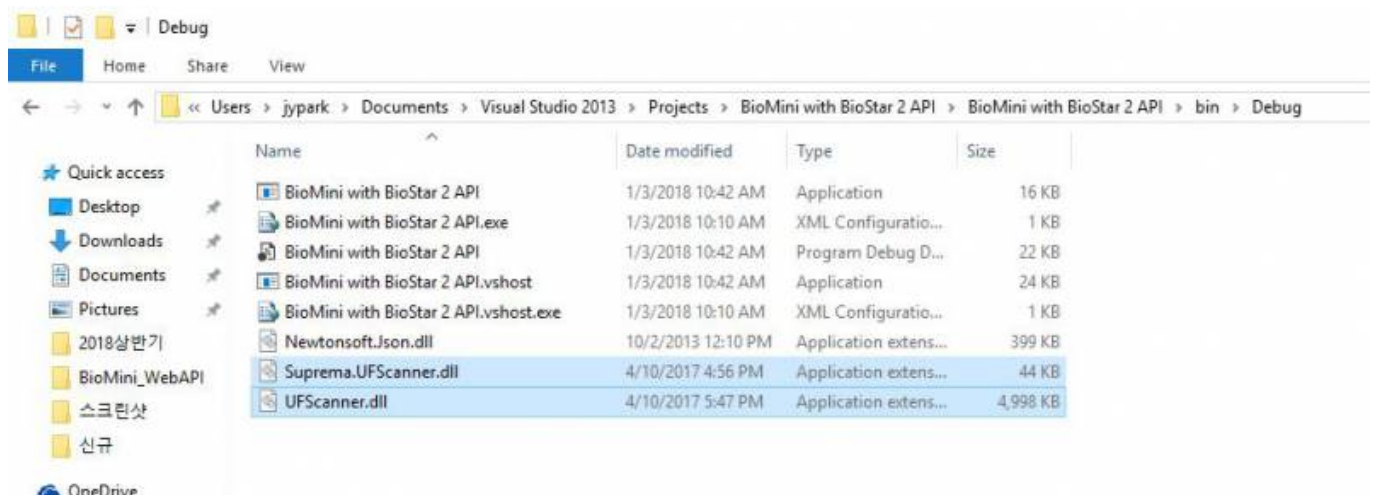
Step 1:

Install BioMini SDK and copy two DLL files in the path (C:\Program Files (x86)\Suprema\BioMini\bin).

- UFSscanner.dll
- Suprema.UFSscanner.dll

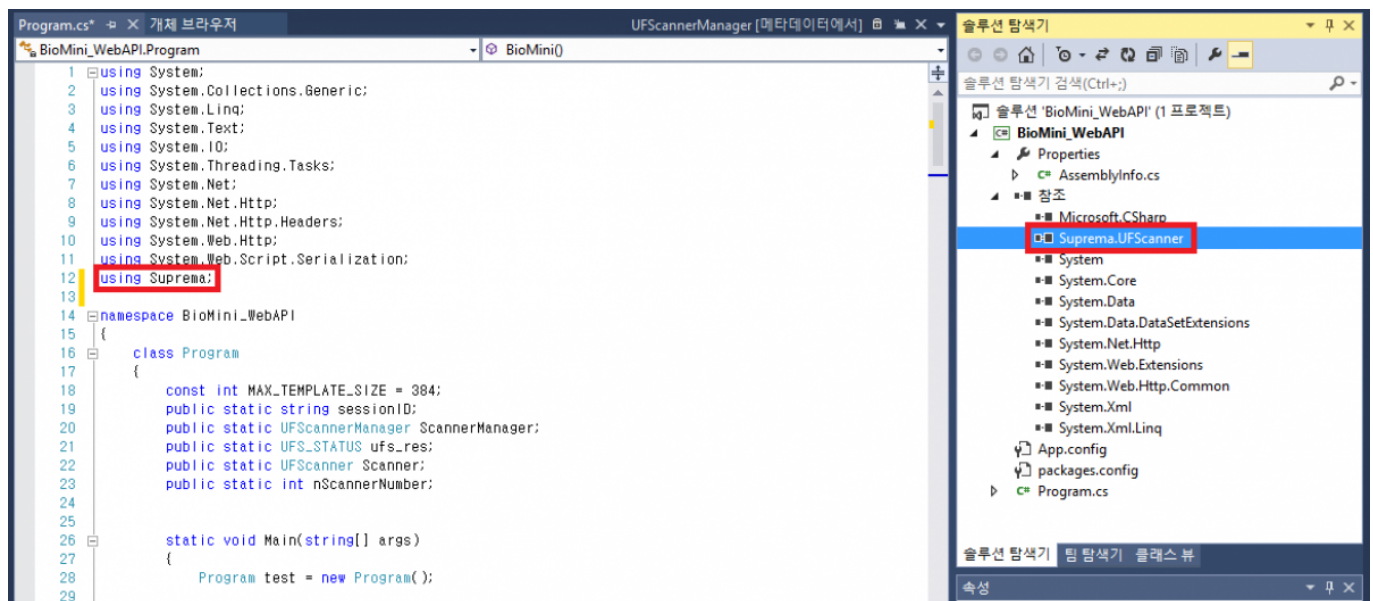


Paste the copied files to your project.



Step 2:

In your project, add the DLL titled **“Suprema.UFScanner”** as reference and add the namespace **Suprema**.



Step 3:

Declare variables and constant. Then, initialize and configure BioMini.



- Line 18, 44: The size of fingerprint template is 384 bytes.

Step 4:

Create new function in order to enroll fingerprint on BioMini.

```

209 static async void EnrollFingerTask()
210 {
211     if (sessionId == null)
212     {
213         Console.WriteLine("You must log in first!");
214         return;
215     }
216
217     CookieContainer cookieContainer = new CookieContainer();
218
219     HttpClientHandler handler = new HttpClientHandler();
220     handler.CookieContainer = cookieContainer;
221
222     HttpClient httpClient = new HttpClient(handler);
223
224     HttpClient client = new HttpClient(handler);
225     cookieContainer.Add(new Uri("http://127.0.0.1:8795"), new Cookie("bs-cloud-session-id", sessionId));
226
227     Console.WriteLine("Input User ID: ");
228     string userInputID = Console.ReadLine();
229
230     string resourceAddress = "http://127.0.0.1:8795/v2/users" + userInputID + "/fingerprint_templates";
231
232     // Enrolling a finger using BioMini
233     byte[] template0 = new byte[MAX_TEMPLATE_SIZE]; // 1st template array
234     byte[] template1 = new byte[MAX_TEMPLATE_SIZE]; // 2nd template array
235     int TemplateSize; // Size of the scanned fingerprint. Returned when calling ExtractEx
236     int EnrollQuality; // Quality of the scanned fingerprint. Returned when calling ExtractEx
237
238     JavaScriptSerializer serializer = new JavaScriptSerializer();
239
240     Dictionary<string, object> dicFinger = new Dictionary<string, object>();
241
242     List<object> fingerprint_template_list = new List<object>();
243     dicFinger.Add("fingerprint_template_list", fingerprint_template_list);
244
245     Console.WriteLine("How many fingerprints do you want to enroll?");
246     int numOffFinger = Convert.ToInt32(Console.ReadLine());
247     for (int idx = 0; idx < numOffFinger; idx++)
248     {
249         Console.WriteLine("Finger #{0}. Scan your finger(1st template scan)", idx + 1);
250         ufs_res = Scanner.ClearCaptureImageBuffer(); // Clearing buffer
251         ufs_res = Scanner.CaptureSingleImage(); // Scans the fingerprint
252         ufs_res = Scanner.ExtractEx(MAX_TEMPLATE_SIZE, template0, out TemplateSize, out EnrollQuality); // Extracts the template from the scanned fingerprint image
253
254         Console.WriteLine("Finger #{0}. Scan your finger(2nd template scan)", idx + 1);
255         ufs_res = Scanner.ClearCaptureImageBuffer();
256         ufs_res = Scanner.CaptureSingleImage();
257         ufs_res = Scanner.ExtractEx(MAX_TEMPLATE_SIZE, template1, out TemplateSize, out EnrollQuality);
258
259         string sTemplate0 = Convert.ToBase64String(template0); // Converts the template data to a Base64 string. When enrolling a finger via API, the template data must be
260         string sTemplate1 = Convert.ToBase64String(template1); // encoded into a Base64 string.
261
262         Dictionary<string, dynamic> dicFingerprintTemplate = new Dictionary<string, dynamic>();
263         fingerprint_template_list.Add(dicFingerprintTemplate);
264         dicFingerprintTemplate["is_prepare_for_duress"] = false;
265         dicFingerprintTemplate["template0"] = sTemplate0;
266         dicFingerprintTemplate["template1"] = sTemplate1;
267     }
268
269     string payload = serializer.Serialize(dicFinger);
270
271     Console.WriteLine("*****[Result of JSON]*****\n");
272     Console.WriteLine(payload);
273
274     StringContent sc = new StringContent(payload, Encoding.UTF8, "application/json");
275     HttpResponseMessage httpResponse = await httpClient.PutAsync(resourceAddress, sc);
276
277     if (httpResponse.IsSuccessStatusCode == true)
278     {
279         Console.WriteLine("User has been created");
280         string httpResponseBody = await httpResponse.Content.ReadAsStringAsync();
281         Console.WriteLine(httpResponseBody);
282     }
283     else
284     {
285         Console.WriteLine("User Creation Failed");
286         Console.WriteLine(httpResponse.ToString());
287     }
288 }

```

- Line 233~236: 1 fingerprint should have 2 templates respectively so that you need to declare two variables for storing each template.
- Line 247~267: Scan Fingerprint with BioMini. (One fingerprint with two templates)

If you would like to get overall sample code for C#, contact us (tech@supremainc.com).

From:

<http://kb.supremainc.com/knowledge/> -

Permanent link:

http://kb.supremainc.com/knowledge/doku.php?id=en:how_to_enroll_fingerprint_on_biomini_via_biostar_2_api

Last update: **2019/04/30 11:14**