MFS100 CLIENT SERVICE WINDOWS - WEB

Ver.: 9.0.3.8

Mantra Softech India Pvt. Ltd.

Version	9.0.3.8
Release Date	14th Feb, 2022
Author	Mr. Rajesh Koriya
Software Support	servico@mantratec.com +91-79-49068000 (Ext : 1)

About:

The document provides the functional and implementation information to work with MFS100 (Mantra Fingerprint Sensor). By using this SDK, you can capture fingerprint from MFS100. This SDK provided facility to extract different-different fingerprint formats like –

- Bitmap Image
- Raw fingerprint image
- ISO (ISO-19794-2/FMR) Template
- ANSI (ANSI-19794-2) Template
- ISO (ISO-19794-4/FIR) Image
- WSQ Image
- Quality of fingerprint
- NFIQ of fingerprint

Dependency:

1. Windows7/8/10 professional or high (home basic or single language is not compatible)
2. VC++ 2008 redistributable (included in MFS100 Driver Setup)
3. Microsoft .Net Framework 4.5.2 or above (must be installed separately)
4. MFS100ClientService setup
5. Mfs100.js
6. jquery-1.8.2.js
7. Port: 8003 (Secure Port) and port 8004 (Non-Secure Port)
8. Url: http://localhost:8004/mfs100/
9. Secure Url: https://localhost:8003/mfs100/

Contents:

Import java script and jquery in web page:

```
<scriptsrc="jquery-1.8.2.js"type="text/javascript"></script>
<scriptsrc="mfs100.js"></script>
```

1. **GetMFS100Info**

   Return: Json Object

   Function is used to get device related information like

   a. Serial Number
   b. Make
   c. Mode
   d. Width of scanner image
   e. Height of scanner image
   f. Local MAC – can be retrieved if network available, may vary if network configuration changed.
   g. Local IP – can be retrieved if network available
   h. System ID – unique system Id
   i. Public IP – can be retrieved only if internet available

```
var res = GetMFS100Info();
if (res.httpStaus) {
document.getElementById('txtStatus').value = "ErrorCode: " + res.data.ErrorCode + "
ErrorDescription: " + res.data.ErrorDescription;

if (res.data.ErrorCode == "0") {
        document.getElementById('tdSerial').innerHTML = res.data.DeviceInfo.SerialNo;
        document.getElementById('tdCertification').innerHTML = res.data.DeviceInfo.Certificate;
        document.getElementById('tdMake').innerHTML = res.data.DeviceInfo.Make;
        document.getElementById('tdModel').innerHTML = res.data.DeviceInfo.Model;
        document.getElementById('tdWidth').innerHTML = res.data.DeviceInfo.Width;
        document.getElementById('tdHeight').innerHTML = res.data.DeviceInfo.Height;
        document.getElementById('tdLocalMac').innerHTML = res.data.DeviceInfo.LocalMac;
        document.getElementById('tdLocalIP').innerHTML = res.data.DeviceInfo.LocalIP;
        document.getElementById('tdSystemID').innerHTML = res.data.DeviceInfo.SystemID;
        document.getElementById('tdPublicIP').innerHTML = res.data.DeviceInfo.PublicIP;
  }
}
else {
   alert(res.err);
}
```

2. **GetMFS100KeyInfo**

   Return: Json Object

   Function is used for locked to get device related information like

   j. Serial Number
   k. Make
   l. Mode
   m. Width of scanner image
   n. Height of scanner image
   o. Local MAC – can be retrieved if network available, may vary if network configuration changed.
   p. Local IP – can be retrieved if network available

q. System ID – unique system Id
r. Public IP – can be retrieved only if internet available

```
var res = GetMFS100Info(String Key); // Base64Encoded Key is for locked sensor
if (res.httpStaus) {
document.getElementById('txtStatus').value = "ErrorCode: " + res.data.ErrorCode + "
ErrorDescription: " + res.data.ErrorDescription;

if (res.data.ErrorCode == "0") {
        document.getElementById('tdSerial').innerHTML = res.data.DeviceInfo.SerialNo;
        document.getElementById('tdCertification').innerHTML = res.data.DeviceInfo.Certificate;
        document.getElementById('tdMake').innerHTML = res.data.DeviceInfo.Make;
        document.getElementById('tdModel').innerHTML = res.data.DeviceInfo.Model;
        document.getElementById('tdWidth').innerHTML = res.data.DeviceInfo.Width;
        document.getElementById('tdHeight').innerHTML = res.data.DeviceInfo.Height;
        document.getElementById('tdLocalMac').innerHTML = res.data.DeviceInfo.LocalMac;
        document.getElementById('tdLocalIP').innerHTML = res.data.DeviceInfo.LocalIP;
        document.getElementById('tdSystemID').innerHTML = res.data.DeviceInfo.SystemID;
        document.getElementById('tdPublicIP').innerHTML = res.data.DeviceInfo.PublicIP;
  }
}
else {
   alert(res.err);
}
```

3. **CaptureFinger(intquality, int timeout)**
Return: Json Object
quality: minimum quality required to successful capture (0 to 100, default = 60)
timeout: maximum milliseconds for capture timeout.

Function is used to capture fingerprint from MFS100.
Different-different fingerprint format and information can be retrieved by this function like bitmapimage, ISO Template (FMR), ISO Image (FIR), Raw fingerprint data, WSQ Image, quality, nfiq etc.

```
var quality = 60; //(1 to 100) (recommended minimum 55)
var timeout = 10; // seconds (minimum=10(recommended), maximum=60, unlimited=0)

var res = CaptureFinger(quality, timeout);
if (res.httpStaus) {
document.getElementById('txtStatus').value = "ErrorCode: " + res.data.ErrorCode + "
ErrorDescription: " + res.data.ErrorDescription;

if (res.data.ErrorCode == "0") {

        document.getElementById('imgFinger').src = "" + res.data.BitmapData;

        var imageinfo = "Quality: " + res.data.Quality + " Nfiq: " + res.data.Nfiq + " W(in):
        " + res.data.InWidth + " H(in): " + res.data.InHeight + " area(in): " +
        res.data.InArea + " Resolution: " + res.data.Resolution + " GrayScale: " +
```

```
        res.data.GrayScale + " Bpp: " + res.data.Bpp + " WSQCompressRatio: " +
        res.data.WSQCompressRatio;

        document.getElementById('txtImageInfo').value = imageinfo;

        document.getElementById('txtIsoTemplate').value = res.data.IsoTemplate;
        document.getElementById('txtAnsiTemplate').value = res.data.AnsiTemplate;
        document.getElementById('txtIsoImage').value = res.data.IsoImage;
        document.getElementById('txtRawData').value = res.data.RawData;
        document.getElementById('txtWsqData').value = res.data.WsqImage;
}
}
else {
  alert(res.err);
}
```

4. CaptureMultiFinger(int quality, int timeout, int nooffinger)

Return: Json Object

quality: minimum quality required to successful capture (0 to 100, default = 60)

timeout: maximum milliseconds for capture timeout.

nooffinger: minimum nooffinger required to successful capture (1 to 10, default = 2)

Function is used to capture fingerprint with de duplication from MFS100.

Captured multiple finger with ISO Template (FMR) and checking de duplication among the captured finger.

```
var quality = 60; //(1 to 100) (recommended minimum 55)
var timeout = 10; // seconds (minimum=10(recommended), maximum=60, unlimited=0)
var nooffinger = 2; // //(1 to 10) (recommended minimum 2)

       var res = CaptureMultiFinger(quality, timeout, nooffinger);

                       if (res.httpStaus) {

                           document.getElementById('txtStatus').value = "ErrorCode: " +
       res.data.ErrorCode + " ErrorDescription: " + res.data.ErrorDescription;

                           if (res.data.ErrorCode == "0") {

                               document.getElementById('txtIsoTemplate').value =
       res.data.IsoTemplate;

                           }
                       }
                       else {
                           alert(res.err);
                   }
```

5. VerifyFinger(string isotemplate1, string isotemplate2)

Return: Json Object

Function is used to match two ISO Template (FMR). For verify ANSI Template please refer MFS100.js file

```
varprobTemplate= document.getElementById('txtIsoTemplate1').value;
vargalleryTemplate= document.getElementById('txtIsoTemplate2').value;

var res = VerifyFinger(probTemplate, galleryTemplate);
// you can pass multiple Template upto maximum 10 by comma separated in galleryTemplate parameter

if (res.httpStaus) {
if (res.data.Status) {
          alert("Finger matched");
     }
else {
if (res.data.ErrorCode != "0") {
               alert(res.data.ErrorDescription);
          }
else {
               alert("Finger not matched");
          }
     }
}
else {
  alert(res.err);
}
```

6. MatchFinger(intquality,int timeout, string isotemplate)

Return: Json Object

Function is used to match finger with current captured finger.For verify ANSI Template please refer MFS100.js file

```
var quality = 60; //(1 to 100) (recommended minimum 55)
var timeout = 10; // seconds (minimum=10(recommended), maximum=60, unlimited=0)

vargalleryTemplate= document.getElementById('txtIsoTemplate').value;
var res = MatchFinger(quality, timeout, galleryTemplate);
// you can pass multiple Template upto maximum 10 by comma separated in galleryTemplate
parameter

if (res.httpStaus) {
if (res.data.Status) {
          alert("Finger matched");
     }
else {
if (res.data.ErrorCode != "0") {
               alert(res.data.ErrorDescription);
        }
else {
             alert("Finger not matched");
          }
    }
}
else {
   alert(res.err);
}
```

7. Biometric Class

Type: Class

```
Biometric(BioType, BiometricData, Pos, Nfiq, Na) {
        this.BioType = BioType;//Valid values “FMR”, “FIR” and “IIR”
        this.BiometricData = BiometricData;//Base64Encoded string of biometric data
        this.Pos = Pos;//Valid values as per UIDAI Specification
        this.Nfiq = Nfiq; // Valid values as per UIDAI Specification. It must be 1 or 2
        this.Na = Na; //Valid values as per UIDAI Specification. It must be 1, 2, 3
}
```

-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-