

suprema
SVP Android SDK
Programming
Interface



Suprema SVP Android SDK Programming Interface User Guide

[Home](#) » [suprema](#) » Suprema SVP Android SDK Programming Interface User Guide 

Contents

- 1 Suprema SVP Android SDK Programming Interface
- 2 Initialize
- 3 Setting Options
- 4 Scan Card
- 5 Setting Fingerprint templates and Fingerprint Identified
- 6 Users finger management (Insert/Update/Delete/Delete all)
- 7 Frequently Asked Questions
- 8 Documents / Resources
 - 8.1 References

suprema

Suprema SVP Android SDK Programming Interface



Initialize

```
package com.example.yourapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Event;
import com.supremainc.sdk.callback.Fingerprint;
import com.supremainc.sdk.callback.Input;
import com.supremainc.sdk.callback.Punch;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    /**
     * DeviceListener receives events about a device.
     */
    private DeviceListener deviceListener = new DeviceListener() {
        @Override
        public void onPunchDetected(Punch data) { }
        @Override
        public void onInputDetected(Input data) { }
        @Override
        public void onEventDetected(Event data) { }
        @Override
        public void onCardScanCompleted(Punch data) { }
        @Override
        public void onFingerprintDetected(Fingerprint data) { }
        @Override
        public void onFingerprintScanCompleted(Fingerprint data) { }
        @Override
        public void onFingerprintIdentified(Fingerprint data) { }
        @Override
        public void onFingerprintScanProgress(int scanTimeout) { }
        @Override
        public void onCardScanProgress(int scanTimeout) { }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

```

        setContentView(R.layout.activity_main);

        /**
         * SDK Initialize.
         */
        svpManager.initialize(this, deviceListener);

        /**
         * SDK service run.
         */
        svpManager.run();
    }

    @Override
    protected void onDestroy() {
        /**
         * SDK service stop.
         */
        svpManager.stop();

        super.onDestroy();
    }
}

```

Setting Options

```

package com.example.yourapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.option.CardOption;
import com.supremainc.sdk.option.FingerprintOption;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() {
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

        setContentView(R.layout.activity_main);

        /**
         * SDK Initialize.
         */
        svpManager.initialize(this, deviceListener);

        /**
         * SDK service run.
         */
        svpManager.run();

        findViewById(R.id.button).setOnClickListener(mClickListener);
    }

    @Override
    protected void onDestroy() {
        /**
         * SDK service stop.
         */
        svpManager.stop();

        super.onDestroy();
    }

    Button.OnClickListener mClickListener = new View.OnClickListener() {
        public void onClick(View v) {
            int result = ;

            /**
             * Card option
             */
            CardOption cardOption = new CardOption();
            cardOption.byteOrder = CardOption.BYTE_ORDER_MSB;
            cardOption.scanTimeout = 5;

            result = svpManager.setCardOption(cardOption);

            /**
             * Fingerprint option
             */
            FingerprintOption fingerOption = new FingerprintOption();
            fingerOption.securityLevel =
FingerprintOption.DEFAULT_SECURITY_LEVEL;
            fingerOption.fastMode = FingerprintOption.DEFAULT_FAST_MODE;
            fingerOption.sensitivity =
FingerprintOption.DEFAULT_SENSOR_SENSITIVITY;
            fingerOption.sensorMode = FingerprintOption.DEFAULT_SENSOR_MODE;
            fingerOption.templateFormat =
FingerprintOption.DEFAULT_TEMPLATE_FORMAT;
            fingerOption.scanTimeout =

```

```

FingerprintOption.DEFAULT_SCAN_TIMEOUT;
    fingerOption.lfdLevel = FingerprintOption.DEFAULT_LFD_LEVEL;
    fingerOption.useAdvancedEnrollment = true;
    fingerOption.useBitmapImage = true;

    result = svpManager.setFingerprintOption(fingerOption);
}
};
}

```

Scan Card

```

package com.example.yourapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Punch;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() {
        /**
         * Called when a card scan completed.
         */
        @Override
        public void onCardScanCompleted(Punch data) {
            Log.i(TAG, "result : " + data.result);
            Log.i(TAG, "card number : " + data.displayString);
        }

        /**
         * Called when a card scan progressed.
         */
        @Override
        public void onCardScanProgress(int scanTimeout) {
            Log.i(TAG, "scanTimeout:" + scanTimeout);
        }
    };

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    /**
     * SDK Initialize.
     */
    svpManager.initialize(this, deviceListener);

    /**
     * SDK service run.
     */
    svpManager.run();

    findViewById(R.id.button).setOnClickListener(mClickListener);
}

@Override
protected void onDestroy() {
    /**
     * SDK service stop.
     */
    svpManager.stop();

    super.onDestroy();
}

Button.OnClickListener mClickListener = new View.OnClickListener() {
    public void onClick(View v) {
        /**
         * Scan card
         */
        int result = svpManager.scanCard();
    }
};
}

```

Scan Finger

```

package com.example.yourapplication;

import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

```

```

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Fingerprint;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() {
        /**
         * Called when a fingerprint scan completed.
         */
        @Override
        public void onFingerprintScanCompleted(Fingerprint data) {
            Log.i(TAG, "result:" + data.result);
            Log.i(TAG, "quality:" + data.quality);

            /**
             * data.template is managed by your application.
             */
            Log.i(TAG, "template:" + data.template.toString());

            if(null != data.image)
            {
                /**
                 * fingerprint bitmap image.
                 */
                Drawable ob = new BitmapDrawable(getResources(),
data.image);
            }

            /**
             * Called when a fingerprint scan progressed.
             */
            @Override
            public void onFingerprintScanProgress(int scanTimeout) {
                Log.i(TAG, "scanTimeout:" + scanTimeout);
            }
        };

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);

            /**
             * SDK Initialize.
             */

```

```

        svpManager.initialize(this, deviceListener);

        /**
         * SDK service run.
         */
        svpManager.run();

        findViewById(R.id.button).setOnClickListener(mClickListener);
    }

    @Override
    protected void onDestroy() {
        /**
         * SDK service stop.
         */
        svpManager.stop();

        super.onDestroy();
    }

    Button.OnClickListener mClickListener = new View.OnClickListener() {
        public void onClick(View v) {
            /**
             * Scan fingerprint.
             */
            int result = svpManager.scanFingerprint();
        }
    };
}

```

Setting Fingerprint templates and Fingerprint Identified

```

package com.example.yourapplication;

import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Fingerprint;
import com.supremainc.sdk.model.Finger;
import com.supremainc.sdk.model.FingerList;
import com.supremainc.sdk.service.DeviceListener;

```

```

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() {
        /**
         * Called when a fingerprint identified.
         */
        @Override
        public void onFingerprintIdentified(Fingerprint data) {
            Log.i(TAG, "result:" + data.result);
            Log.i(TAG, "id:" + data.id);
            Log.i(TAG, "templateSize:" + data.templateSize);
            Log.i(TAG, "quality:" + data.quality);
            Log.i(TAG, "fingerprintIndex:" + data.fingerprintIndex);
            Log.i(TAG, "isUpdated:" + data.isUpdated);

            if (null != data.image)
            {
                Drawable ob = new BitmapDrawable(getResources(),
data.image);
            }
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        /**
         * SDK Initialize.
         */
        svpManager.initialize(this, deviceListener);

        /**
         * SDK service run.
         */
        svpManager.run();

        findViewById(R.id.button).setOnClickListener(mClickListener);
    }

    @Override
    protected void onDestroy() {
        /**
         * SDK service stop.
         */
        svpManager.stop();
    }
}

```

```

        super.onDestroy();
    }

    Button.OnClickListener mClickListener = new View.OnClickListener() {
        public void onClick(View v) {

            /**
             * Fingerprint templates are managed by your application.
             * Template array represents the data received as
             "onFingerprintScanCompleted"
             */
            byte[][] template = new
byte[Finger.TEMPLATE_PER_FINGER][Fingerprint.FINGERPRINT_TEMPLATE_SIZE];

            FingerList fingerList = new FingerList();
            for(int i = 0; i < Finger.MAX_NUM_OF_FINGER; i++)
            {
                Finger finger = new Finger();
                finger.id = i;
                finger.index = (int)(Math.random() *
FingerList.MAX_NUM_OF_FINGER_PER_USER);

                finger.setTemplate(0, template[0]);
                finger.setTemplate(1, template[1]);

                fingerList.addFinger(finger);
            }

            /**
             * Fingerprint matching is done with the set templates.
             * If the fingerprint matches successfully, the
             "onFingerprintIdentified" is called.
             */
            int result = svpManager.setFingerList(fingerList);
        }
    };
}

```

Users finger management (Insert/Update/Delete/Delete all)

```

package com.example.yourapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

```

```

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Fingerprint;
import com.supremainc.sdk.model.Finger;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() {
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        /**
         * SDK Initialize.
         */
        svpManager.initialize(this, deviceListener);

        /**
         * SDK service run.
         */
        svpManager.run();

        findViewById(R.id.button).setOnClickListener(mClickListener);
    }

    @Override
    protected void onDestroy() {
        /**
         * SDK service stop.
         */
        svpManager.stop();

        super.onDestroy();
    }

    Button.OnClickListener mClickListener = new View.OnClickListener() {
        public void onClick(View v) {
            int result = ;
            /**
             * Fingerprint templates are managed by your application.
             * Template array represents the data received as
             "onFingerprintScanCompleted"
             */
            byte[][] template = new
byte[Finger.TEMPLATE_PER_FINGER][Fingerprint.FINGERPRINT_TEMPLATE_SIZE];

```

```

        int id = 1000;

        Finger finger = new Finger();
        finger.id = id;
        finger.index = ;
        finger.setTemplate(, template[]);
        finger.setTemplate(1, template[1]);

        /**
         * Insert new finger.
         */
        result = svpManager.insertFinger(finger);

        /**
         * Update new finger.
         */
        result = svpManager.updateFinger(finger);

        /**
         * Delete finger.
         */
        result = svpManager.deleteFinger(finger);

        /**
         * Delete all fingers.
         */
        result = svpManager.deleteAllFinger();
    }
};
}

```

Data detected (Card/Finger/Input)

```

package com.example.yourapplication;

import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Fingerprint;
import com.supremainc.sdk.callback.Input;
import com.supremainc.sdk.callback.Punch;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

```

```

SvpManager svpManager = new SvpManager();
private static final String TAG = "YourApp";

private DeviceListener deviceListener = new DeviceListener() {
    /**
     * Called when a card detected.
     */
    @Override
    public void onPunchDetected(Punch data) {
        Log.i(TAG, "result : " + data.result);
        Log.i(TAG, "card number : " + data.displayString);
    }

    /**
     * Called when a finger detected.
     */
    @Override
    public void onFingerprintDetected(Fingerprint data) {
        Log.i(TAG, "result:" + data.result);
        Log.i(TAG, "template:" + data.template.toString());
        Log.i(TAG, "quality:" + data.quality);

        if (null != data.image)
        {
            /**
             * fingerprint bitmap image.
             */
            Drawable ob = new BitmapDrawable(getResources(),
data.image);
        }
    }

    /**
     * Called when an input(TTL/Tamper) detected.
     */
    @Override
    public void onInputDetected(Input data) {
        Log.i(TAG, "result : " + data.result);
        Log.i(TAG, "type : " + data.type);
        Log.i(TAG, "port : " + data.port);
        Log.i(TAG, "status : " + data.status);

        if (data.type == Input.INPUT_TYPE_TTL) {
            Log.i(TAG, "TTL input detected.");
        }
        else if (data.type == Input.INPUT_TYPE_TAMPER) {
            Log.i(TAG, "Tamper input detected.");
        }
    }
};

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    /**
     * SDK Initialize.
     */
    svpManager.initialize(this, deviceListener);

    /**
     * SDK service run.
     */
    svpManager.run();
}

@Override
protected void onDestroy() {
    /**
     * SDK service stop.
     */
    svpManager.stop();

    super.onDestroy();
}
}

```

LED / Output Control

```

package com.example.yourapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.define.Channel;
import com.supremainc.sdk.define.LedColor;
import com.supremainc.sdk.define.Relay;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() { };
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    /**
     * SDK Initialize.
     */
    svpManager.initialize(this, deviceListener);

    /**
     * SDK service run.
     */
    svpManager.run();

    findViewById(R.id.button).setOnClickListener(mClickListener);
}

@Override
protected void onDestroy() {
    /**
     * SDK service stop.
     */
    svpManager.stop();

    super.onDestroy();
}

Button.OnClickListener mClickListener = new View.OnClickListener() {
    public void onClick(View v) {
        int result = ;
        /**
         * Execute LED action.
         */
        result = svpManager.executeLedAction(LedColor.LED_COLOR_RED);
        /**
         * Execute output action.
         */
        result = svpManager.executeOutputAction(Channel.RELAY_PORT_0,
Relay.ON);
    }
};
}

```

Firmware upgrade

```

package com.example.yourapplication;

import android.support.v7.app.AppCompatActivity;

```

```

import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.define.Channel;
import com.supremainc.sdk.define.LedColor;
import com.supremainc.sdk.define.Relay;
import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    private DeviceListener deviceListener = new DeviceListener() {
        /**
         * Called when a device event detected.
         */
        @Override
        public void onEventDetected(Event data) {
            Log.i(TAG, "result : " + data.result);
            Log.i(TAG, "code : " + data.code);

            if( data.code == EventCode.EVENT_DEVICE_FIRMWARE_UPGRADED )
            {
                /**
                 * Reboot device.
                 */
                if (data.result == ErrorCode.SUCCESS ) {
                    svpManager.rebootDevice();
                }
            }
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        /**
         * SDK Initialize.
         */
        svpManager.initialize(this, deviceListener);

        /**
         * SDK service run.
         */
        svpManager.run();
    }
}

```

```

        findViewById(R.id.button).setOnClickListener(mClickListener);
    }

    @Override
    protected void onDestroy() {
        /**
         * SDK service stop.
         */
        svpManager.stop();

        super.onDestroy();
    }

    public void ftpUpgrade()
    {
        int result = ;
        /**
         * FTP firmware option.
         */
        FirmwareOption option = new FirmwareOption();
        option.type = FirmwareOption.FTP_UPGRADE;
        option.fileName = "omnis_1.0.0_20181201.bin";
        option.host = "127.0.0.1";
        option.port = 21;
        option.username = "suprema";
        option.password = "suprema";

        /**
         * Connect FTP server.
         */
        result = svpManager.connectFtpServer(option);

        /**
         * Get firmware file list.
         */
        ArrayList<String> fileList = new ArrayList<String>();
        result = mManager.getFirmwareFileList(fileList);

        /**
         * Upgrade firmware.
         */
        result = svpManager.upgradeFirmware(option);

        /**
         * Disconnect FTP server.
         */
        result = svpManager.disconnectFtpServer();
    }

    public void usbUpgrade()
    {

```

```

        int result = ;
        /**
         * USB firmware option.
         */
        FirmwareOption option = new FirmwareOption();
        option.type = FirmwareOption.USB_UPGRADE;
        option.fileName = "omnis_1.0.0_20181201.bin";

        /**
         * Upgrade firmware.
         */
        result = svpManager.upgradeFirmware(option);
    }

    Button.OnClickListener mClickListener = new View.OnClickListener() {
        public void onClick(View v) {

            /**
             * FTP firmware upgrade.
             */
            ftpUpgrade();

            /**
             * USB firmware upgrade.
             */
            usbUpgrade();
        }
    };
}

```

Ethernet Setting

```

package com.example.yourapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import android.support.v7.app.AppCompatActivity;

import com.supremainc.sdk.SvpManager;
import com.supremainc.sdk.callback.Event;
import com.supremainc.sdk.callback.Fingerprint;
import com.supremainc.sdk.callback.Input;
import com.supremainc.sdk.callback.Punch;
import com.supremainc.sdk.define.RequestCode;

```

```

import com.supremainc.sdk.service.DeviceListener;

public class MainActivity extends AppCompatActivity {

    SvpManager svpManager = new SvpManager();
    private static final String TAG = "YourApp";

    // DeviceListener receives events about a device.
    private DeviceListener deviceListener = new DeviceListener() {
        @Override
        public void onPunchDetected(Punch data) { }
        @Override
        public void onInputDetected(Input data) { }
        @Override
        public void onEventDetected(Event data) { }
        @Override
        public void onCardScanCompleted(Punch data) { }
        @Override
        public void onFingerprintDetected(Fingerprint data) { }
        @Override
        public void onFingerprintScanCompleted(Fingerprint data) { }
        @Override
        public void onFingerprintIdentified(Fingerprint data) { }
        @Override
        public void onFingerprintScanProgress(int scanTimeout) { }
        @Override
        public void onCardScanProgress(int scanTimeout) { }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // SDK Initialize.
        svpManager.initialize(this, deviceListener);

        // SDK service run.
        svpManager.run();

        findViewById(R.id.button).setOnClickListener(mClickListener);
    }

    @Override
    protected void onDestroy() {
        // SDK service stop.
        svpManager.stop();

        super.onDestroy();
    }
}

```

```

        Button.OnClickListener mClickListener = new View.OnClickListener() {
            public void onClick(View v) {
                boolean useDHCP = false;
                String ip = "192.168.1.123";
                String subnet = "255.255.255.0";
                String gateway = "192.168.1.1";
                String dns = "192.168.1.100";

                svpManager.setEthernetConfig(useDHCP, ip, subnet, gateway, dns);
            }
        };

        @Override
        protected void onActivityResult(int requestCode, int resultCode, Intent data) {
            super.onActivityResult(requestCode, resultCode, data);

            if (requestCode == RequestCode.ETHERNET_REQUEST_CODE) {
                if (resultCode == RESULT_OK) {
                    int result = data.getIntExtra("result", -1);
                    Toast.makeText(this, "Result: " + result,
                        Toast.LENGTH_SHORT).show();
                }
                else {
                    Toast.makeText(this, "Failed", Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
}

```

From

- <https://kb.supremainc.com/svpsdk/-SVPAndroidSDK>

Permanent link:

https://kb.supremainc.com/svpsdk/doku.php?id=en:quick_guide

- Last update: 2019/09/20 11:10

Frequently Asked Questions

Q: What is the SVP Android SDK?

A: The SVP Android SDK is a software development kit provided by Suprema Inc. for integrating fingerprint and card scanning features into Android applications.

Q: How do I initialize the SDK in my Android application?

A: To initialize the SDK, follow the steps outlined in the user manual

1. Import necessary packages.
2. Create an instance of SvpManager.
3. Implement DeviceListener to handle device events.
4. Call svpManager.initialize() with the context and device listener.
5. Start the SDK service by calling svpManager.run().

Documents / Resources

	Suprema SVP Android SDK Programming Interface [pdf] User Guide SVP Android SDK Programming Interface, Android SDK Programming Interface, SDK Programming Interface, Programming Interface
---	--

References

-  [DATA ID: One simple link you only need.](#)
-  [SVP Android SDK \[SVP Android SDK \]](#)
-  [Quick Guide \[SVP Android SDK \]](#)
- [User Manual](#)

[Manuals+.](#) [Privacy Policy](#)

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.