UIDAI

Unique Identification Authority of India

Planning Commission, Govt. of India (GoI),

3rd Floor, Tower II,

Jeevan Bharati Building,

Connaught Circus,

New Delhi 110001

Biometric Fingerprint Sensor Integration API

Draft

Version 0.93

Contents

1 Introduction

The Unique Identification Authority of India (UIDAI) has been created, with the mandate of providing a unique identity to all Indian residents. The UIDAI proposes to use biometrics also to conduct authentication transactions in future. Proof of concept studies is being conducted in various phases in order to ascertain state of technology as well as viability of the concept.
Fingerprint sensors are vital components required for carrying out fingerprint authentication. As discussed in various concept papers, UIDAI proposes to offer biometric authentication through combination of UID number + biometric. Initially, fingerprint based biometric authentication would be supported.

2 Objective of the document

The current version of this document has been provided for feedback from the Application developers, as well as Biometric fingerprint Capture Device manufacturers. This API is intended for fingerprint Biometric device manufacturers to publish API. This API will facilitate smooth integration of the API into the UIDAI POC application.

3 Application Overview

The API is specified as communication protocol between the UID POC application Software and the Fingerprint Biometric Capture Devices, The UID POC application. The proposed interface does not address any security-related issues. After the security requirements are defined, the specifications may need to be modified to address them.

UID biometric POC Application software is being developed on Java Platform. POC application is designed to accept UID number of the resident, complete the fingerprint capture process and then transmit the data in encrypted packet as per the UIDAI authentication specification. Following functions are expected to be performed by the UIDAI POC application.

1. Accept the UID number from the resident (No API calls required here)
2. Choose one among many sensors connected to the PC through USB. (init() will be called)
3. Proceed to collect the biometric solution from the resident.
    a. Fingerprint Raw Image as per ISO format (ISO 19794-4) (captureSample() will be called)
    b. Fingerprint Minutiae (Template) as per ISO format (ISO 19794-2) (extractTemplate() will be called)
    c. BMP of Fingerprint Image in order to display the image in the application interface. (createImage() will be called)
4. In cases where more than one fingerprint is collected, application is expected to compare the captured fingerprints with already captured fingerprints and verify for duplication. (compareTemplates() will be called)
5. Form the authentication request as per the Authentication API and send it to server and store the image and minutiae in the log files of the device for future analysis.
6. All buffered transactions will be stored in table in mysql db in the system

4 Installer Design

All applications are expected to provide a Installers that installs the libraries and other required components on the system. Initial expectation is that the installer be available on windows XP and above platform.

In addition installing the libraries, the interface API as per specification is expected to be installed in the following directory as per the naming convention provided below.

Please expect to find the **FingerprintSensors** Folder in '**C**' directory of every system. Within the **FingerprintSensors** Directory, installer has to create a directory with naming convention listed below.

DeviceManufacturer- DeviceModelIdentifier-DeviceVendor

For Example :
Device manufacturer: XYZ
Device Vendor – Abc - Device vendor who has responded to EOI.
Device Model: Fingerprint 3000 series

Here, the DeviceManufacturer tag is "XYZ" and Device Model Identifier is "Fingerprint 3000 series". Hence the folder name in the destination name **could** be
XYZ- FINGERPRINT3000-Abc.

C:/FingerprintSensors/XYZ-FINGERPRINT3000-Abc

Within this folder, the installer is expected to have the following files.

1. A DLL file which implements the API as per the specification cited above. The naming convention for single point of access – **UIDAIFingerprintSensorAPI.DLL**.
    1. If more than one dll is required for any sensor, for example as in case of a sensor with a different extractor matcher, then needs to add all other dll files in the same folder. All API calls should be wrapped in one dll file **UIDAIFingerprintSensorAPI.DLL**.
2. Configuration or property files(if any) which customizes the output of the DLL. Text files are preferred. This will help UIDAI teams also to customize the output of the sensor.
3. Readme Document – that explains the property files and any other important information related to the API and configuration.

5 API Specifications

The development platform: JAVA 1.6, Windows XP or above

The below Interface API should be implemented by the device vendors for their supporting devices.

The Interface to be implemented by the device vendors:

AuthPOC2FingerPrintAPI.java

//Begin of Fingerprint API

```java
package in.gov.uidai.auth.biometric;

import java.awt.Image;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Properties;

/**
 * The below Interface should be implemented by the device vendors for their
 * supporting fingerprint devices.
 *
 */
public class AuthPOC2FingerPrintAPI {

/**
 * Return codes.
 * All API calls should return 0 for SUCCESS.
 */
public static final int FINGERPRINT_API_DEVICE_INIT_SUCCESSFUL = 0;
public static final int FINGERPRINT_API_DEVICE_INIT_UNSUCCESSFUL = -1;
public static final int FINGERPRINT_API_DEVICE_ERR_UNABLE_TO_CAPTURE = -2;
public static final int FINGERPRINT_API_DEVICE_ERR_TOO_MANY_HANDLES = -3;
public static final int FINGERPRINT_API_DEVICE_ERR_TIMEOUT_EXPIRED = -4;
public static final int FINGERPRINT_API_DEVICE_ERR_PURPOSE_NOT_SUPPORTED = -5;
public static final int FINGERPRINT_API_TEMPLATES_MATCH = 6;
public static final int FINGERPRINT_API_TEMPLATES_NOTMATCH = -6;
public static final int FINGERPRINT_API_DEVICE_CAPTURE_SUCCESSFUL = 10;
public static final int FINGERPRINT_API_DEVICE_CAPTURE_UNSUCCESSFUL = -10;
public static final int FINGERPRINT_API_DEVICE_PURPOSE_VERIFICATION = 12;
public static final int FINGERPRINT_API_DEVICE_PURPOSE_ENROLLMENT = 13;
public static final int FINGERPRINT_API_TEMPLATE_ISO_FORMAT = 16;
public static final int FINGERPRINT_API_TEMPLATE_ANSI_FORMAT = 17;
public static final int FINGERPRINT_API_DEVICE_SAMPLE_QUALITY_GOOD = 18;
public static final int FINGERPRINT_API_DEVICE_SAMPLE_QUALITY_POOR = -18;
public static final int FINGERPRINT_API_TEMPLATE_EXTRACT_UNSUCCESSFUL = -20;
public static final int FINGERPRINT_API_TEMPLATE_EXTRACT_SUCCESSFUL = 20;
public static final int FINGERPRINT_API_DEVICE_INIT_DLL_ALREADY_LOADED = -21;
public static final boolean FINGERPRINT_API_DEVICE_CONNECTED = true;
public static final boolean FINGERPRINT_API_DEVICE_DISCONNECTED = false;
public static final int FINGERPRINT_API_DEVICE_CLOSE_SUCCESSFUL = 22;
public static final int FINGERPRINT_API_DEVICE_CLOSE_UNSUCCESSFUL = -22;
public static final String FINGERPRINT_API_DEVICE_IMAGE_BMP_FORMAT = "bmp";

public static String dllDirectory = null;
public static String dllFile = null;
private static Properties props = new Properties();
private static final String CONF_FILE = "conf/config.properties";
private static FileInputStream in = null;

//Constructor to load the dll
public AuthPOC2FingerPrintAPI(String dllVendor) {

        try {
                in = new FileInputStream(CONF_FILE);
        } catch (FileNotFoundException e) {
                System.out.println("Conf file error: " + e.getMessage());
        }
        try {
                props.load(in);
                in.close();
        } catch (IOException e) {
            System.out.println("Conf file loading error: " + e.getMessage());
        }
        dllDirectory = props.getProperty("uidai.authpoc2.dll.dir");
        dllFile = props.getProperty("uidai.authpoc2.dll");
        System.out.println("java.library.path : "
                + System.getProperty("java.library.path"));
        try {
            File f = new File(dllDirectory + "/" + dllVendor);
            if (f.isDirectory()) {
                if (f.list().length < 1) {
                    System.out.println("Folder is empty.");
                    return;
                }
            }
            System.load(dllDirectory + "/" + dllVendor + "/" + dllFile);
            System.out.println("Loaded: " + dllDirectory + "/" + dllVendor
                    + "/" + dllFile);
        } catch (java.lang.UnsatisfiedLinkError e) {
            System.out.println(e.getMessage());
        }
}

        //Device initialization call
        public native int init();

        //Device connection check
        public native boolean isDeviceConnected();

        //Capture sample call returns the Sample object. Please see below
        //Sample.java for more info
        public native Sample captureSample(int purpose, int timeout);

        //Extract template call returns the Template object. Please see below
        //Template.java for more info
        public native Template extractTemplate(Sample sample, int format);

        //Compare Templates call returns the MatchResult object. Please see below
        //MatchResult.java for more info
        public native MatchResult compareTemplates(byte[] tData1, byte[] tData2);

        //Create Image call returns the Image for the format specified, bmp,
        //jpeg or jp2 or etc
        /**
         * Hard code the height and width from the captured sample as of now.
         */
        public native Image createImage(byte[] rawImage, String format);

        //Device info call returns the DeviceInfo object. Please see below
        //DeviceInfo.java for more info
        public native DeviceInfo getDeviceInfo();

        //Extractor info call returns Extractor object. Please see below
        //ExtractorInfo.java for more info
        public native ExtractorInfo getExtractorInfo();

        //Close call returns the device connection closure.
        public native int close();


        //Not a mandatory api.
        /**
         * Returns All data in one call. Not a mandatory API.
         * @param purpose : Enrollment
         * @param timeout : 10000ms
         * @param isoFormat1 : ISO19794-4 (FIR)
         * @param isoFormat2 : ISO19794-2 (FMR)
         * @param imageFormat : .bmp
```

```
	 * @return
	 */
	public native FingerPrintData getAllData(int purpose, int timeout,
			int isoFormat1, int isoFormat2, String imageFormat);

	//Below is not a API
	public void finalize() {
		System.out.println("\nAPI class is garbage collected.");
	}
}
```

//End of Fingerprint API

Sample.java

```
package in.gov.uidai.auth.biometric;

public class Sample {

	byte[] rawImage = null;
	byte[] iso_19794_2_Template = null; // FMR
	byte[] iso_19794_4_Image = null; // FIR

	int rawImageSize = 0;
	int rawImageWidth = 0;
	int rawImageHeight = 0;
	int isoTemplateSize = 0; //FMR size
	int isoRawImageSize = 0; //FIR size
	int nfiq = 0; //NFIQ Score - Mandatory
	int sampleQuality =
	AuthPOC2FingerPrintAPI.FINGERPRINT_API_DEVICE_SAMPLE_QUALITY_POOR;
	String errorString = null;
	int errorCode = 0;

	//New Field is added
	//This is the “Time” taken in "milliseconds" to
	//capture a fingerprint by the device, i.e. the time in milliseconds
	//between "Finger is Pressed and Released" for a fingerprint capture
	long fpCaptureTime = -1L;


	//New methods added to access the fpCaptureTime data
	/**
	 * @return the fpCaptureTime
	 */
	public long getFpCaptureTime(){
		return fpCaptureTime;
	}

	/**
	 * @param fpCaptureTime the fpCaptureTime to set
	 */
	public void setFpCaptureTime(long fpCaptureTime){
		this.fpCaptureTime = fpCaptureTime;
	}


	/**
	 * @return the nfiq - Values(1,2,3,4,5)-Mandatory
	 */
	public int getNfiq() {
		return nfiq;
	}

	/**
	 * @param nfiq the nfiq to set
	 */
	public void setNfiq(int nfiq) {
		this.nfiq = nfiq;
	}

	/**
	 * @return the isoRawImageSize
	 */
	public int getIsoRawImageSize() {
		return isoRawImageSize;
	}

	/**
	 * @param isoRawImageSize the isoRawImageSize to set
	 */
	public void setIsoRawImageSize(int isoRawImageSize) {
		this.isoRawImageSize = isoRawImageSize;
	}

	/**
	 * @return the rawImage
	 */
	public byte[] getRawImage() {
		return rawImage;
	}

	/**
	 * @param rawImage
	 *            the rawImage to set
	 */
	public void setRawImage(byte[] rawImage) {
		this.rawImage = rawImage;
	}

	/**
	 * @return the iso_19794_2_Template
	 */
	public byte[] getIso_19794_2_Template() {
		return iso_19794_2_Template;
	}

	/**
	 * @param iso_19794_2Template
	 *            the iso_19794_2_Template to set
	 */
	public void setIso_19794_2_Template(byte[] iso_19794_2Template) {
		iso_19794_2_Template = iso_19794_2Template;
	}

	/**
	 * @return the iso_19794_4_Image
	 */
	public byte[] getIso_19794_4_Image() {
		return iso_19794_4_Image;
	}

	/**
	 * @param iso_19794_4Image
	 *            the iso_19794_4_Image to set
	 */
	public void setIso_19794_4_Image(byte[] iso_19794_4Image) {
		iso_19794_4_Image = iso_19794_4Image;
	}
	/**
	 * @return the rawImageSize
	 */
	public int getRawImageSize() {
		return rawImageSize;
	}

	/**
	 * @param rawImageSize
	 *            the rawImageSize to set
	 */
	public void setRawImageSize(int rawImageSize) {
		this.rawImageSize = rawImageSize;
	}

	/**
	 * @return the rawImageWidth
	 */
	public int getRawImageWidth() {
		return rawImageWidth;
	}

	/**
	 * @param rawImageWidth
	 *            the rawImageWidth to set
	 */
	public void setRawImageWidth(int rawImageWidth) {
		this.rawImageWidth = rawImageWidth;
	}

	/**
	 * @return the rawImageHeight
	 */
	public int getRawImageHeight() {
		return rawImageHeight;
	}

	/**
	 * @param rawImageHeight
	 *            the rawImageHeight to set
	 */
	public void setRawImageHeight(int rawImageHeight) {
		this.rawImageHeight = rawImageHeight;
	}

	/**
	 * @return the isoTemplateSize
	 */
	public int getIsoTemplateSize() {
		return isoTemplateSize;
	}

	/**
	 * @param isoTemplateSize
	 *            the isoTemplateSize to set
	 */
	public void setIsoTemplateSize(int isoTemplateSize) {
		this.isoTemplateSize = isoTemplateSize;
	}

	/**
	 * @return the sampleQuality
	 */
	public int getSampleQuality() {
		return sampleQuality;
	}
```

```
    /**
     * @param sampleQuality
     *            the sampleQuality to set
     */
    public void setSampleQuality(int sampleQuality) {
        this.sampleQuality = sampleQuality;
    }

    /**
     * @return the errorString
     */
    public String getErrorString() {
        return errorString;
    }

    /**
     * @param errorString
     *            the errorString to set
     */
    public void setErrorString(String errorString) {
        this.errorString = errorString;
    }

    /**
     * @return the errorCode
     */
    public int getErrorCode() {
        return errorCode;
    }

    /**
     * @param errorCode
     *            the errorCode to set
     */
    public void setErrorCode(int errorCode) {
        this.errorCode = errorCode;
    }

}
```

Template.java

```
package in.gov.uidai.auth.biometric;

public class Template {
    byte[] iso_19794_2_Template = null;
    int isoTemplateSize = 0;

    String errorString = null;
    int errorCode = 0;

    /**
     * @return the iso_19794_2_Template
     */
    public byte[] getIso_19794_2_Template() {
        return iso_19794_2_Template;
    }

    /**
     * @param iso_19794_2Template
     *            the iso_19794_2_Template to set
     */
    public void setIso_19794_2_Template(byte[] iso_19794_2Template) {
        iso_19794_2_Template = iso_19794_2Template;
```

```java
        }

        /**
         * @return the isoTemplateSize
         */
        public int getIsoTemplateSize() {
                return isoTemplateSize;
        }

        /**
         * @param isoTemplateSize
         *            the isoTemplateSize to set
         */
        public void setIsoTemplateSize(int isoTemplateSize) {
                this.isoTemplateSize = isoTemplateSize;
        }

        /**
         * @return the errorString
         */
        public String getErrorString() {
                return errorString;
        }

        /**
         * @param errorString
         *            the errorString to set
         */
        public void setErrorString(String errorString) {
                this.errorString = errorString;
        }

        /**
         * @return the errorCode
         */
        public int getErrorCode() {
                return errorCode;
        }

        /**
         * @param errorCode
         *            the errorCode to set
         */
        public void setErrorCode(int errorCode) {
                this.errorCode = errorCode;
        }
}
```

DeviceInfo.java

```java
package in.gov.uidai.auth.biometric;

public class DeviceInfo {

        String make = null;
        String model = null;
        String serialNumber = null;

        String errorString = null;
        int errorCode = 0;

        /**
         * @return the make
         */
        public String getMake() {
                return make;
        }
```

```
	/**
	 * @param make
	 *            the make to set
	 */
	public void setMake(String make) {
		this.make = make;
	}

	/**
	 * @return the model
	 */
	public String getModel() {
		return model;
	}

	/**
	 * @param model
	 *            the model to set
	 */
	public void setModel(String model) {
		this.model = model;
	}

	/**
	 * @return the serialNumber
	 */
	public String getSerialNumber() {
		return serialNumber;
	}

	/**
	 * @param serialNumber
	 *            the serialNumber to set
	 */
	public void setSerialNumber(String serialNumber) {
		this.serialNumber = serialNumber;
	}

	/**
	 * @return the errorString
	 */
	public String getErrorString() {
		return errorString;
	}

	/**
	 * @param errorString
	 *            the errorString to set
	 */
	public void setErrorString(String errorString) {
		this.errorString = errorString;
	}

	/**
	 * @return the errorCode
	 */
	public int getErrorCode() {
		return errorCode;
	}

	/**
	 * @param errorCode
	 *            the errorCode to set
	 */
	public void setErrorCode(int errorCode) {
		this.errorCode = errorCode;
```

```
        }
}
```

Extractor.java

```
package in.gov.uidai.auth.biometric;

public class ExtractorInfo {

        String vendor = null;
        String name = null;
        String version = null;

        String errorString = null;
        int errorCode = 0;

        /**
         * @return the vendor
         */
        public String getVendor() {
                return vendor;
        }

        /**
         * @param vendor
         *              the vendor to set
         */
        public void setVendor(String vendor) {
                this.vendor = vendor;
        }

        /**
         * @return the name
         */
        public String getName() {
                return name;
        }

        /**
         * @param name
         *              the name to set
         */
        public void setName(String name) {
                this.name = name;
        }

        /**
         * @return the version
         */
        public String getVersion() {
                return version;
        }

        /**
         * @param version
         *              the version to set
         */
        public void setVersion(String version) {
                this.version = version;
        }

        /**
         * @return the errorString
         */
        public String getErrorString() {
                return errorString;
        }
```

```
	/**
	 * @param errorString
	 *            the errorString to set
	 */
	public void setErrorString(String errorString) {
		this.errorString = errorString;
	}

	/**
	 * @return the errorCode
	 */
	public int getErrorCode() {
		return errorCode;
	}

	/**
	 * @param errorCode
	 *            the errorCode to set
	 */
	public void setErrorCode(int errorCode) {
		this.errorCode = errorCode;
	}

}
```

MatchResult.java

```
package in.gov.uidai.auth.biometric;

public class MatchResult {
	boolean matchFound = false;
	int matchScore = 0;

	String errorString = null;
	int errorCode = 0;

	/**
	 * @return the mathFound
	 */
	public boolean isMatchFound() {
		return matchFound;
	}

	/**
	 * @param mathFound
	 *            the mathFound to set
	 */
	public void setMatchFound(boolean matchFound) {
		this.matchFound = matchFound;
	}

	/**
	 * @return the matchScore
	 */
	public int getMatchScore() {
		return matchScore;
	}

	/**
	 * @param matchScore
	 *            the matchScore to set
	 */
	public void setMatchScore(int matchScore) {
		this.matchScore = matchScore;
	}
	/**
	 * @return the errorString
	 */
	public String getErrorString() {
		return errorString;
	}

	/**
	 * @param errorString
	 *            the errorString to set
	 */
	public void setErrorString(String errorString) {
		this.errorString = errorString;
	}

	/**
	 * @return the errorCode
	 */
	public int getErrorCode() {
		return errorCode;
	}

	/**
	 * @param errorCode
	 *            the errorCode to set
	 */
	public void setErrorCode(int errorCode) {
		this.errorCode = errorCode;
	}

}
```

Note:
The getAllData() is not a mandatory api to be implemented.

FingerPrintData.java

```
package in.gov.uidai.auth.biometric;

import java.awt.Image;

public class FingerPrintData {

	byte[] rawImage = null;
	byte[] iso_19794_2_Template = null; // FMR
	byte[] iso_19794_4_Image = null; // FIR
	int rawImageSize = 0;
	int rawImageWidth = 0;
	int rawImageHeight = 0;
	int isoTemplateSize = 0;
	int sampleQuality = 0;

	String make = null;
	String model = null;
	String serialNumber = null;

	String vendor = null;
	String name = null;
	String version = null;

	Image image = null;

	String errorString = null;
	int retCode = 0;

	/**
	 * @return the rawImage
	 */
	public byte[] getRawImage() {
		return rawImage;
	}

	/**
	 * @param rawImage
	 *            the rawImage to set
	 */
	public void setRawImage(byte[] rawImage) {
		this.rawImage = rawImage;
	}

	/**
	 * @return the iso_19794_2_Template
	 */
	public byte[] getIso_19794_2_Template() {
		return iso_19794_2_Template;
	}

	/**
	 * @param iso_19794_2Template
	 *            the iso_19794_2_Template to set
	 */
	public void setIso_19794_2_Template(byte[] iso_19794_2Template) {
		iso_19794_2_Template = iso_19794_2Template;
	}

	/**
	 * @return the iso_19794_4_Image
	 */
	public byte[] getIso_19794_4_Image() {
		return iso_19794_4_Image;
	}

	/**
	 * @param iso_19794_4Image
	 *            the iso_19794_4_Image to set
	 */
	public void setIso_19794_4_Image(byte[] iso_19794_4Image) {
		iso_19794_4_Image = iso_19794_4Image;
	}

	/**
	 * @return the rawImageSize
	 */
	public int getRawImageSize() {
		return rawImageSize;
	}

	/**
	 * @param rawImageSize
	 *            the rawImageSize to set
	 */
	public void setRawImageSize(int rawImageSize) {
		this.rawImageSize = rawImageSize;
	}

	/**
	 * @return the rawImageWidth
	 */
	public int getRawImageWidth() {
		return rawImageWidth;
	}

	/**
	 * @param rawImageWidth
	 *            the rawImageWidth to set
 */
public void setRawImageWidth(int rawImageWidth) {
	this.rawImageWidth = rawImageWidth;
}

/**
 * @return the rawImageHeight
 */
public int getRawImageHeight() {
	return rawImageHeight;
}

/**
 * @param rawImageHeight
 *            the rawImageHeight to set
 */
public void setRawImageHeight(int rawImageHeight) {
	this.rawImageHeight = rawImageHeight;
}

/**
 * @return the isoTemplateSize
 */
public int getIsoTemplateSize() {
	return isoTemplateSize;
}

/**
 * @param isoTemplateSize
 *            the isoTemplateSize to set
 */
public void setIsoTemplateSize(int isoTemplateSize) {
	this.isoTemplateSize = isoTemplateSize;
}

/**
 * @return the sampleQuality
 */
public int getSampleQuality() {
	return sampleQuality;
}

/**
 * @param sampleQuality
 *            the sampleQuality to set
 */
public void setSampleQuality(int sampleQuality) {
	this.sampleQuality = sampleQuality;
}

/**
 * @return the make
 */
public String getMake() {
	return make;
}

/**
 * @param make
 *            the make to set
 */
public void setMake(String make) {
	this.make = make;
}

/**
 * @return the model
```

```
	 */
	public String getModel() {
		return model;
	}

	/**
	 * @param model
	 *            the model to set
	 */
	public void setModel(String model) {
		this.model = model;
	}

	/**
	 * @return the serialNumber
	 */
	public String getSerialNumber() {
		return serialNumber;
	}

	/**
	 * @param serialNumber
	 *            the serialNumber to set
	 */
	public void setSerialNumber(String serialNumber) {
		this.serialNumber = serialNumber;
	}

	/**
	 * @return the vendor
	 */
	public String getVendor() {
		return vendor;
	}

	/**
	 * @param vendor
	 *            the vendor to set
	 */
	public void setVendor(String vendor) {
		this.vendor = vendor;
	}

	/**
	 * @return the name
	 */
	public String getName() {
		return name;
	}

	/**
	 * @param name
	 *            the name to set
	 */
	public void setName(String name) {
		this.name = name;
	}

	/**
	 * @return the version
	 */
	public String getVersion() {
		return version;
	}

	/**
	 * @param version
	 *            the version to set
	 */
	public void setVersion(String version) {
		this.version = version;
	}

	/**
	 * @return the image
	 */
	public Image getImage() {
		return image;
	}

	/**
	 * @param image
	 *            the image to set
	 */
	public void setImage(Image image) {
		this.image = image;
	}

	/**
	 * @return the errorString
	 */
	public String getErrorString() {
		return errorString;
	}

	/**
	 * @param errorString
	 *            the errorString to set
	 */
	public void setErrorString(String errorString) {
		this.errorString = errorString;
	}

	/**
	 * @return the retCode
	 */
	public int getRetCode() {
		return retCode;
	}

	/**
	 * @param retCode
	 *            the retCode to set
	 */
	public void setRetCode(int retCode) {
		this.retCode = retCode;
	}
}
```

End of API document.