

# Information Security

## ID Device

## Software Development Kit (SDK) with ActiveX Controls (Option)

## Version 1.9

## Brief Description

**Note:** Contact details for further information on the use of the SDK are as follows:

SIEMENS ICM RDC IS BIO

E-Mail: [fingertip@mch.siemens.de](mailto:fingertip@mch.siemens.de)

Europe: <http://www.fingertip.de/produkte/idmousesdk.asp?lang=eng>

Japan: <http://www.siemens.co.jp/idmouse>

North America: <http://www.siemensidmouse.com>

**Published by SIEMENS ICM RDC IS BIO PM**  
Hofmannstr. 51, D - 81379 Munich

## Preface

This description of the ID Device SDK is aimed at all users, developers and integrators of biometric systems who want to integrate the Siemens/Cherry ID/ST products as a fingerprint device in their applications. Knowledge of the Windows 98, Windows 2000, Windows NT 4.0 (not supported by the ST device !) or Windows XP operating system is required for the use of the ID Device SDK. Additional knowledge of biometrics is not needed.

The ID Device SDK 1.90 is independent from the connected USB device and supports the following ID Devices only by checking their device identifiers.

- Siemens ID Mouse A10
- Siemens ID Mouse professional
- Cherry ID Keyboard
- ST Microelectronics TouchChip® Fingerprint Reader

The users preselection prior SDK installation decides whether the drivers for the Siemens ID Mouse, for the Cherry ID Keyboard or for the ST TouchChip® Fingerprint Reader must to be installed from the SDK CD Rom.

## Short description

The ID Device SDK is designed for integrating the biometric recognition mechanisms of the Siemens ID technology in the most varied software applications.

It contains the Siemens fingerprint recognition functions for extending customer applications with regard to increased security and userfriendliness .

In the simplest scenario, conventional passwords and PINs can be accessed and retrieved using finger reference data for identification and verification purposes.

The ID Device SDK includes functions for feeding in a person's details into the system (enrolment), for updating and improving reference data in the archive, for managing archive and individual user data as well as basic functions for identifying and verifying a person.

Because of the modularity of the ID Device SDK, its possible range of uses is extremely varied. We can distinguish two groups of users. On the one hand, the system integrators, who integrate the ID Device SDK in their own systems and create their own applications. These are typical Windows programmers who do not require special biometric knowledge. The biometric integrators form the second group. Their aim is to integrate the ID Device SDK and its biometric functions in order to supplement or expand their security applications.

The FAPI function library is reentrant. Multiple parallel running applications using the ID Devices are possible. The functions of the ID Device SDK are invoked via standard C function interfaces.

An integrator may use the Siemens ID Mouse or the Cherry ID Keyboard as a capture device for fingerprints via the ID Device SDK. Captured raw data can be converted to windows bitmap format. Based on this image data other image and verification algorithms than those provided with the ID Device SDK can be used. If an integrator decides to do so all other ID Device SDK support will be lost, not only for the biometric functions like enrollment, verification and identification but also for archive management. In this case the integrator takes over the responsibility for the performance of the biometric system.

## Relationship between biometry and biometric algorithms of the ID Device SDK

The Siemens ID Device SDK allows simple enhancement of existing applications with biometric security. Biometry, in general, refers to the checking of a person's identity using that person's unique personal features. These biometric features can be derived from physiological procedures like facial or fingerprint recognition or from behavioral procedures like speech or dynamic signature recognition. In contrast to identification features, such as keys or smartcards, biometric features cannot be lost, stolen or used by unauthorized persons.

There are basically three processes in each biometric system:

### 1. Enrolment

A person's reference data is generated in the enrolment process. The reference data contains the most basic information about a person's biometric features. In the case of the ID Device SDK, these are the finger-specific features. During the downstream identification and verification processes in the biometric system, this reference data is used as a basis for comparison with the current features. Each person is requested to present the selected finger three times for the feeding-in process.

### 2. Verification

Verification means checking the person id against the predefined identity, i.e. the statement "I am" is verified. This means that the identity of the person to be compared must be known before the start of the verification process. This can be done, for example, by specifying the person's name or a user ID or using a chipcard.

### 3. Identification

Identification means that the biometric system checks the identity of the person comparing it with all persons known to the system. In other words, the identity of the person to be checked is not specified before the identification commences, rather the identity is returned as a result of a successful identification.

The highly simplified diagram below shows the general scheme for the fingerprint recognition algorithm. The main components here are the encoder and the matcher. The encoder is responsible for extracting the finger-specific features, while the matcher is responsible for comparing the current finger-specific (query) features with those in the stored reference.

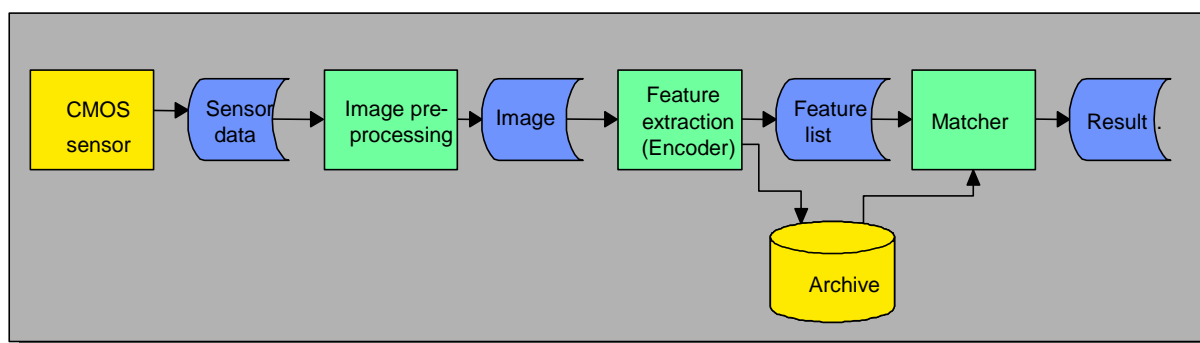


Figure1: Process flow in ID Device SDK

### **Scope of the ID Device SDK**

The ID Device SDK comprises the following three components:

1. CD ROM ID Device SDK with:
  - Siemens Demonstration Suite
  - Sample code
  - Installation program
  - User documentation
2. Siemens ID Mouse, Cherry ID Keyboard or ST TouchChip<sup>®</sup> Fingerprint Reader
3. CD-ROM ID Device SW 4.0 (optional) with:
  - Installation program
  - User documentation

The ID Device SDK is a statically loadable Dynamic Link Library (DLL) with defined C interfaces. The demo program gives the user an initial impression in graphic form of the functionality of the ID Device SDK. The demo program is easy to use and is self-explanatory. The sample code contains typical calls of all necessary ID Device SDK functions.

The software integrator is provided with a User Guide and Programmer's Guide for using the ID Device SDK.

Users can develop and market their own applications based on the ID Device SDK.

### **Startup ID Device SDK 1.9**

#### **System requirements**

Please note the following requirements for using the ID Device SDK and for developing biometric applications with the ID Device SDK as well as for invoking the demo program:

#### **Hardware requirements**

- PC with INTEL Pentium processor from 233 MHz, Monitor with minimum resolution of 800 x 600 pixels, at least 64 MByte RAM
- USB port
- Siemens ID Mouse, Cherry ID Keyboard, ST TouchChip<sup>®</sup> Fingerprint Reader

#### **Software requirements**

- Operating system: Windows 98, Windows 2000 or Windows NT 4.0 as SP3 or higher
- ID Device SDK 1.90 software package installed
- Microsoft Visual Studio 6.0 (or other compatible tools) as development tool

## Installation process

1. Connect the Siemens ID Mouse, Cherry ID Keyboard or ST TouchChip® Fingerprint Reader to the USB port on the PC.
2. Set the color resolution of the screen to more than 256 colors and color resolution to at least 600 x 800 pixels.
3. Place the ID Device SDK installation CD in the CD-ROM drive and start "Setup.exe". Then follow the installation instructions. The Setup program installs the ID Device SDK in the selected destination directory. Perform a reboot if necessary.

## Architecture of the ID Device SDK

The architecture diagram below depicts the relationship of the ID Device SDK with customer applications and the Siemens ID Mouse. The internal component structure is also shown in simple form. Also the Cherry ID Keyboard can be connected as the ID Device.

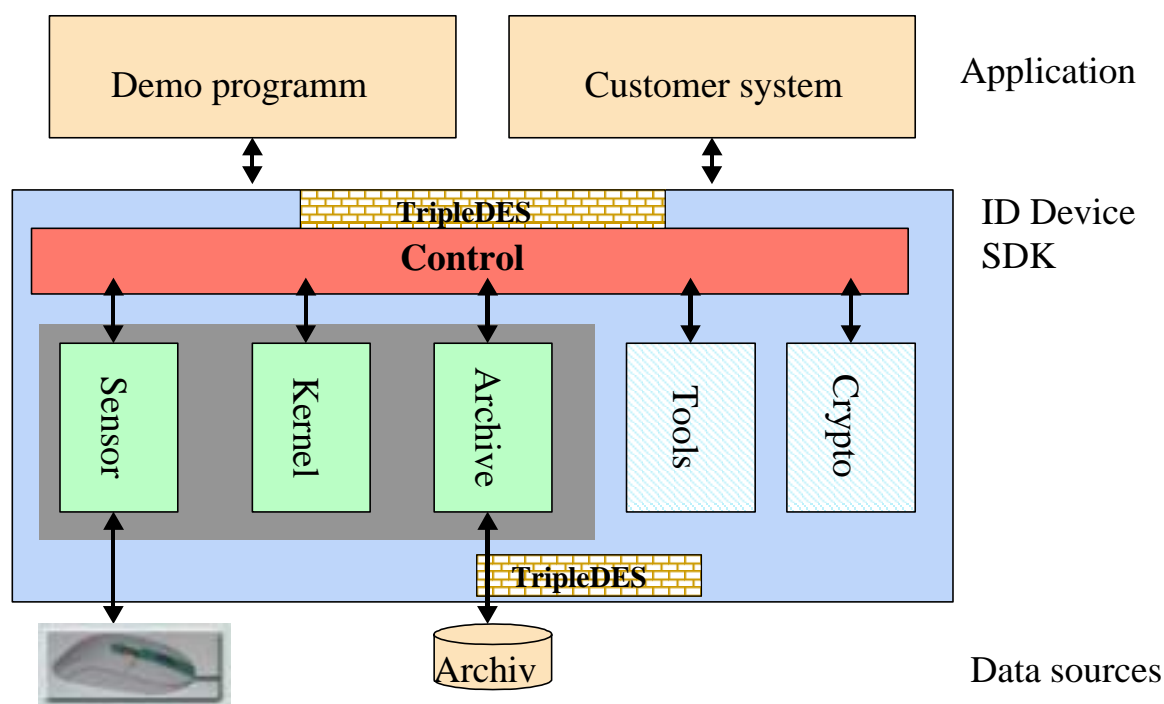


Figure 2: Architecture of the ID Device SDK

- **"Control" component**  
This component allows control of all ID Device SDK V1.90 functions and provides the functional interfaces for external communication with the entire application.
- **"Kernel" component**  
This component contains all basic functions for processing fingerprints.
- **"Sensor" component**  
This component is used for controlling the Sensor hardware within the Siemens ID Mouse or the Cherry ID Keyboard and provides a fully imported image with corrections from the FingerTIP™ sensor chip.

- **"Archive" component**  
This component contains functions for storing, deleting and reading user data or biometric data from an archive.
- **"Tools" component**  
This component provides help and administration functions that can be used by the other components.
- **"Crypto" component**  
This component provides encryption and decryption functions as well as functions for creating signatures for the reference data stored in the ID Device SDK archive.

## **Functionality of the ID Device SDK**

### **Sensor Management functions**

These functions are used for activating and testing the FingerTip™ sensor, which is integrated in the appropriate ID Device. They control communication with the sensor and enable the loading of raw fingerprint images, which are used for displaying or producing reference data. These functions can be used independently from the biometric functions.

### **Biometric functions**

This group of functions forms the cornerstone of the ID Device SDK and includes creation of biometric data from raw fingerprint images (encoder) and comparison of this biometric data with reference entries in the archive (matcher). These basic functions are the basis for identifying or verifying a person.

The creation and improvement of finger reference data can be performed either by encoding the raw image imported by the sensor or by copying over bitmap files from external databases with subsequent processing in the SDK.

The verification of a person is supported in two forms. Firstly the verification is performed (1:1 comparison) by correlating the current finger reference with a reference entry in the SDK archive, which is addressed uniquely from a variety of archive entries on the basis of a finger ID. Secondly the reference data for comparison is also transferred by the application when the verification process is invoked (e.g. read from a smartcard) and compared with the current fingerprint image. No internal SDK archive is necessary in this case since the reference data is managed by the application and made available to the SDK as required using appropriate function calls. Because of the modularity of the SDK, reference data for fingers can be stored as required on smartcards and within client/server platforms and retrieved from there by the application. This means that a variety of smartcard applications or client/server applications can be implemented individually for customers.

### **Archive Management functions**

This group contains all function calls required for managing the archive, for example deletion of finger entries, specific and sequential reading of personal data and modification of specific user data (passwords, etc.).

**Comment:**

The ID Device SDK does not offer person management. Reference data managed in the ID Device SDK archive is finger-oriented, i.e. a finger ID is returned following corresponding recognition. The application must establish the relationship between the stored reference data (finger) and the associated person.

**Help and management functions**

This group of functions provides help functions for reading or storing bitmap files as well as functions for converting reference data (biodata) to predefined data formats.

Other functions are available, for example for requesting memory releases or for returning error clarification texts based on specific error numbers.

**Encryption**

The reference data stored in the archive and also the biometric data transferred through the FAPI functions is encrypted with TripleDES (128-bit key length) and signed. Biometric data is no more readable or can be faked by any other application even when sending the biometric data via network to a server or when storing it on a smartcard. The customer application is responsible for the transfer of the biometric data between client and server as well as for the storage on server or on a smartcard.

**Relationship with biometric standards**

The ID Device SDK is based on the Human Authentication API (HA-API) V2.0 and Biometric API (BAPI) V1.1 biometric standards.

**Compatibility to earlier versions**

Program sources using earlier SDK versions 1.5/1.6 /1.8 can use the ID Device SDK 1.9 with small or no changes, because known functions do not change their FAPI calls. New SDK functions have to be supported by the application through function calls, which have to be implemented. Old fingerprint archives from the SDK 1.6 can not be used anymore, because of the added device authorisation, new crypto-mechanisms and algorithms-upgrade for recognition, that means that a new enrolment procedure is required for the users.

If the ID Mouse SDK 1.5/1.6/1.8 based application accessed registry keys the registry path string must be changed from "...\\ID Mouse SDK\\1.50..." to "...\\ ID Device SDK 1.90...".

**New functions/enhancements of the ID Device SDK 1.9 in comparison to older versions (1.6/1.8)**

- (a) USB drivers for Siemens ID Mouse A10/Professional, Cherry ID Keyboard and ST Microelectronics TouchChip<sup>®</sup> Fingerprint Reader (ST TCRS1 Smart Card Reader) available.
- (b) ID converter program as a separate tool to convert templates/BIO data retrieved from the ID Devices (Mouse, Keyboard, ST) to BIO dataformat which can be down loaded to the Siemens embedded solution TopSec ID Module or Siemens Matching on Card Technology (Smart Card Technology) and vice versa for all directions.
- (d) Live matching functionality for feedback messages to support intelligent user guidance during sensor readout (image capturing) which results in further enhancements of the recognition performance.

- (e) Multiple sensor device support, which means that several devices (Mouse, Keyboard or ST) can be connected to one PC.
- (f) Improvement of the product's security features
- (g) Support for Windows XP operating system
- (h) Free adjustment of the biometric threshold for advanced biometric applications
- (i) Support of the Siemens "Matching on Card" technology
- (j) Integration of the Siemens FT Technology Version 11.0 with increased algorithm performance.
- (k) Improvement of the enrolment feedback
  - (i) Prematcher function for shorter matching time

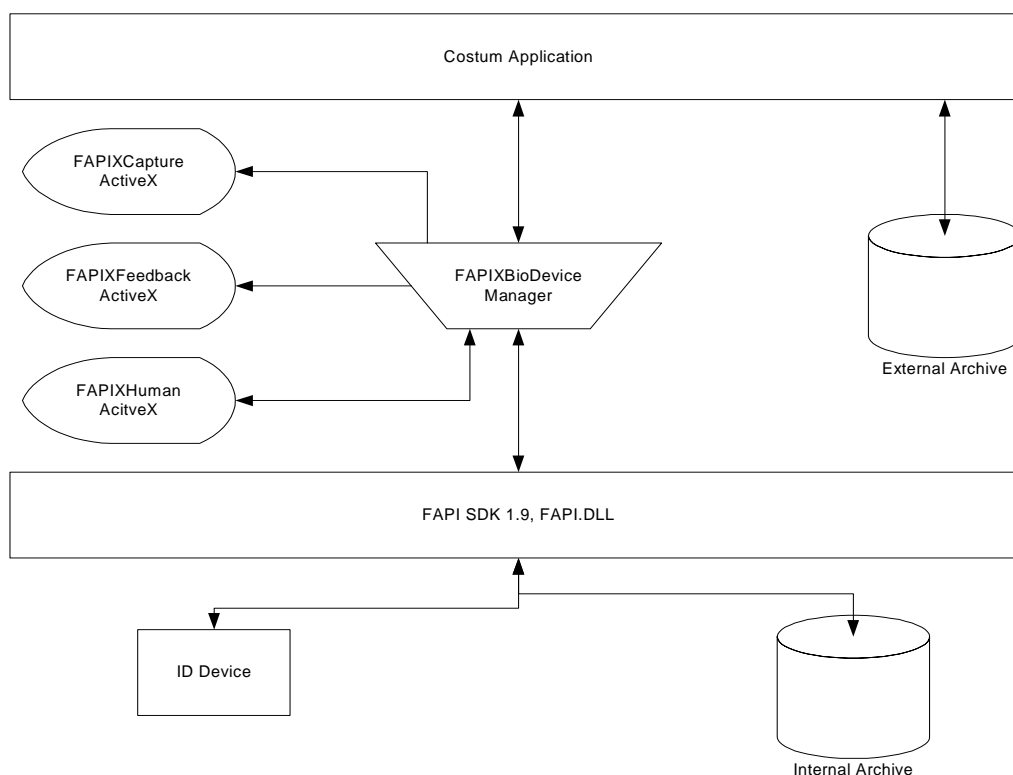
### Additional software for the ID Device SDK 1.9 (option)

#### ActiveX controls fore easy integration and advanced "look and feel"

The ActiveX controls, based on the SDK 1.9, enable an easy integration into software applications and adaption of the "look and feel" of the ID Mouse professional software. German and English language versions are available.

The ActiveX DLL covers the whole functionality of the FAPI.DLL. This component is registered to the system during the setup.

Figure 3 shows the basic architecture of the ActiveX.



**Figure 3: Architecture of the ID Device SDK 1.9/ ActiveX**

### ActiveX component – FAPIXCapture Control

This component evaluates the messages provided by the FapiXBioDeviceManager and is exclusively used to graph the picture taken by the ID Sensor. Background image (e.g. company logo) and dimension of this control can be changed by the properties of the object. Additionally a textbox on the bottom of the control can be used to show information about the picture. The control also provides a function to save the captured image to a bitmap file.



Figure 4: FAPIXCapture Control

### ActiveX component – FAPIXFeedback Control

This ActiveX control serves to evaluate the feedback information during the enrolment/capture process and to show animations (“Put your finger on sensor”, ...) and other images (see Siemens Biometrics Demonstration Suite).



Figure 5: FAPIXFeedback Control

## ActiveX component – FAPIXHuman

This component serves to select the fingers for the enrolment process and to send the information to the FAPIXBioDeviceManager or to show the information provided by the FAPIXBioDeviceManager. Enrolled fingers are labelled with a green button and the currently selected finger is marked with a blue arrow. Optionally the fingers can also be labelled with a red button (“manipulated”). The control also provides a textbox to show information to the currently selected finger in English or German.

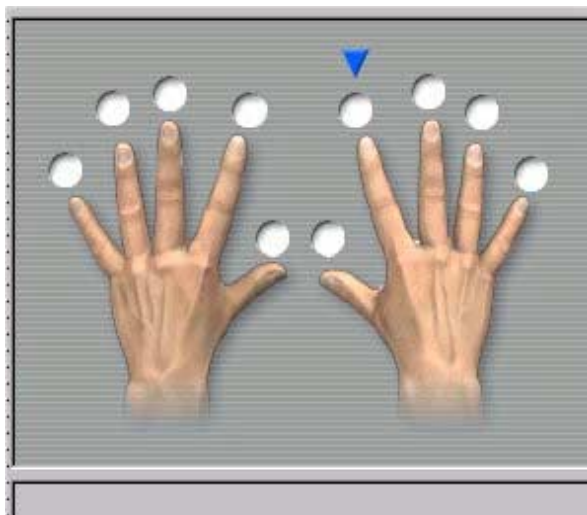


Figure 6: FAPIXHuman Control

## Sample Code: Verify – Identify - Enrolment

This sample code will be used to show the verification/identification/enrolment process and can be used for the application programming. It contains all other ActiveX SDK 1.9 controls.

## Hardware and Software Requirements:

### Hardware Requirements

- min. PC Pentium II 233 Mhz
- min. 64 MB RAM
- USB port
- CD Rom

Windows NT, 2000, XP:

- Siemens ID Mouse inclusive FingerTIP I sensor
- Siemens ID Mouse Professional inclusive FingerTIP I sensor
- Cherry FingerTIP ID Board inclusive FingerTIP I sensor

Windows 2000, XP:

- ST Microelectronics TCRS1A Touch Chip Device

**Note:** Windows NT not supported by the ST device

### **Software Requirements**

- Operating systems:
  - Windows 2000 SP2 or higher
  - Windows NT 4.0 SP6 or higher (ST devices not supported)
  - Windows XP
- Adobe Acrobat Reader 4.0 or higher (for documentation only)

### **Reference material:**

[1] ID Device SDK User's Guide

[2] ID Device SDK Programmer's Guide

[3] Data Book FingerTIP™ CMOS Chip and System, Infineon Technologies AG

[4] ID Device SDK / ActiveX Programmer's Guide

### **Support:**

The price for the SDK includes 4 hours integration – and SDK –support.