DigitalPersona™
Software Development Kit
Gold Edition 2.1.0

DigitalPersona's Gold Software Development Kit (SDK) enables developers to add the power of fingerprint authentication security to their Windows applications. The toolkit includes header files that define the API, sample code for Visual C++ 6.0, and the Fingerprint Recognition libraries. For customers who use Visual Basic to develop programs it will be necessary to use the DigitalPersona Platinum SDK or to write a wrapper to package the Gold SDK DLLs for their use. The SDK does not come with a U.are.U Fingerprint sensor or license to the DigitalPersona Fingerprint Engine. You will need to purchase a sensor and license separately.

Who it is for?

A system integrator can use the SDK to easily add the functionality of authenticating a user with fingerprint recognition to his/her software. The SDK can be used to develop a wide variety of custom applications related to PC or client/server access, time and attendance, and physical access.

What is the SDK not for?

The SDK is designed to develop applications that run on the Windows platform. If you have PC applications running on a different operating system, please contact us. Also, the standard shipping version of the SDK only works with DigitalPersona sensors. Custom SDKs using the core DigitalPersona Fingerprint Recognition technology can be provided to run with other fingerprint sensors.

What is the architecture of the SDK and the DigitalPersona Fingerprint Engine?

There are several components to the DigitalPersona Fingerprint Engine. The application developer is provided with two levels of APIs, the low-level feature extraction and matching interface, and the high-level authentication functions.

Fingerprint Engine Components

The Feature Extraction Module:
The feature extraction module extracts a template from the fingerprint image that comes from the sensor. The standard feature extraction module works with fingerprint images produced with the DigitalPersona sensors. The size of the template is approximately 300 bytes. The application has the flexibility to use a smaller template size to meet stringent memory requirements for storing the template. The feature extraction process takes approximately 0.6 seconds on a Pentium 200. The feature extraction module also returns diagnostic information about the quality of the fingerprint captured, such as poor contrast, or blurred image.

The Matching Module :
The matching module takes two fingerprint templates, performs a match, and verifies that the fingerprints come from the same finger. The match process takes approximately 0.1 seconds on a Pentium 200. The application can set the security level (False Accept Rate) to be used for matches. The default security setting is 1.5% chance of false reject vs. 0.01% chance of false accept. If a match is confirmed, the matching module returns an l28-bit string, which can be used as a unique, reproducible key for that user. The matching module is also able to perform learning upon successful match. Learning requires extra computation and can be switched on and off by the application.

The Database Module:
The database module can be used to store a database of user attributes and fingerprint templates. It is included as a convenience for developers. Fingerprint templates can be stored anywhere the developer chooses, such as in an extension to a digital certificate, within a LDAP directory, in the NT SAM user database, or on a smart card. The user record contains multiple fingerprints of the user, and a protected storage area where user data such as passwords can be securely stored. The protected storage area is made available only after an authentication of the user through a correct fingerprint match. The database files are protected against unauthorized changes. Authentication is required for database updates. The "user session" functionality is provided to avoid repeated authentication.

The High Level Interface Module:
This module exports high-level functions for registration and verification. The Engine does not implement its own user interface. Instead, it performs callbacks to functions provided by the developer, so that consistency of application user interface can be achieved. This module interacts with the DigitalPersona sensor server to simplify the image acquisition and the device event monitoring. Most developers will need to only use this module.

Sensor Server
The sensor server is a COM Server that handles multiple sensors connected to the USB ports. It dispatches notification of fingerprint capture and device events to the client applications running on the system. The Server sets up a challenge/response encrypted link with the sensor to securely transfer an encrypted image.

Security and privacy measures:
DigitalPersona has given great consideration to security and user privacy issues. The sensor server sets up a challenge-response encrypted link with the sensor to securely transfer the image. Fingerprint templates are always returned encrypted from the Recognition System, and user records are stored encrypted in the database. Furthermore, registration templates cannot be matched against themselves — which means the user must provide a current new sample for matching. A security key is entered by the user during installation, and is used internally as part of the internal encryption schema of the Fingerprint Engine. Templates that are created on computers using different security keys are not compatible. Please note that the encryption functionality is not exported by the toolkit.

System Requirements

- CD-ROM drive
- Pentium-class processor (or better)
- 16Mbytes of RAM
- Universal Serial Bus (USB) Port on computer where the sensor is connected
- Windows XP, Me, 2000, NT 4.0 and 98