A DigitalPersona Whitepaper

Best Practices for Implementing Fingerprint Biometrics in Applications

Tips and guidelines for achieving high performance in fingerprint-enabled applications using DigitalPersona's U.are.U® family of SDKs

September 2012

Biometrics can help you enhance the security and usability of your application. By following a few simple guidelines and using DigitalPersona's biometric software development kits, you can easily add fast fingerprint identification and verification capabilities that enable your application to recognize individual users without requiring other forms of ID. This can be used in a variety of ways – from sign-on and confirmation of important actions to special approvals by other users – to help combat fraud and boost customer efficiency.

Introduction

Fingerprint biometrics makes it fast and easy for your application to determine who is using it. Biometrics can be used to:

- Identify users without requiring other forms of ID (such as usernames, ID numbers or swipe cards).
- Verify another form of identification without requiring passwords or PINs.
- Confirm that particular actions are being performed by the right user – turning the fingerprint sensor into a kind of "Enter" key that tells your application who is doing what.
- Prevent unauthorized access and stop former users from sneaking into your application.

This whitepaper provides a variety of guidelines and tips that can help you use fingerprint biometrics to boost the security and usability of your application. It complements the documentation provided for DigitalPersona's U.are.U® software development kits.

Benefits of Using Fingerprint Biometrics

Fingerprints provide a compelling way to differentiate your application:

- **Accountability** – Fingerprints can tie actions to specific individuals – deterring inappropriate behavior.
- **Workforce Management** – Fingerprints provide accurate time and attendance tracking, reducing waste.
- **Loss Prevention** – Supervisors' fingerprints can be required for special actions, facilitating adherence to corporate policies.
- **Compliance** – Fingerprints can be used to provide an audit trail identifying who came in contact with sensitive data.

Keys to a Successful Application

Applications that use fingerprint biometrics most successfully often have the following attributes:

- **Simple setup** – Your application should guide users through registering or "enrolling" their fingerprint, typically when a user account is added. This usually takes about a minute and is only done once, often in the presence of a supervisor or administrator.

- **Ease of use** – With a few visual cues from your application, fingerprints can be used with little effort to link specific actions to the individuals who perform them.

- **Speed** – Implemented correctly, fingerprints can be used to recognize individuals within groups of thousands of people in under a second.

- **Flexibility** – Allow users to register whichever fingers are most convenient for them and allow two or more fingers to be used so that there is a backup in case of injury.

- **Privacy** – Always store and use Fingerprint Minutiae Data (FMD), not raw images. This is much more efficient and helps protect users' privacy.

- **Logging** – Record all uses of – and failures to use – biometrics, including details such as time, place, and context within your application and so on.

Important Concepts About Fingerprints

"Biometrics" literally means the measuring of a person's physical traits. It is a technology that can be used to recognize and authenticate individuals based on who they are, instead of what they know (passwords or PINs) or what they possess (keys or swipe cards).

There are many types of biometrics, including palm or iris scanning, voice and face recognition. Fingerprints are the most widely used form of biometrics in commercial applications. Fingerprint sensors are now built into most notebook computers, are offered as an option on many brands of point of sale (POS) stations, and are increasingly being used in door locks, medical dispensary cabinets and other embedded devices.

When adding fingerprints to your application, the most important concepts to understand are:

- **Fingerprints are unique** – No two people, even identical twins, have the same fingerprints.

- **Everybody has fingerprints** – But, sometimes the prints on one or more fingers can become difficult to read. Rough physical labor can wear prints down and dry skin (whether due to climate or constant washing with alcohol-based cleaners) can make prints harder to detect. In contrast, body oil on fingers can actually help make fingerprints easier to read.

- **Images** and **Fingerprint Minutiae Data (FMD)** – When a user touches a fingerprint sensor, the hardware scans the pad of their finger to capture an image of their fingerprint. Commercial applications rarely use or store the raw fingerprint images; instead, they convert the image into an object containing Fingerprint Minutiae Data (FMD), a much smaller mathematical representation of a fingerprint image and then discard the original image. Fingerprint Minutiae Data cannot be converted back into the original image.

- **Registration** or **Enrollment** – Scanning a person's fingerprints the first time is called registration or enrollment. This is typically done by an application in a controlled, secure setting, often under the supervision of an attendant. During enrollment, it is common practice to capture multiple scans of a fingerprint to increase accuracy and so that people can later touch the fingerprint sensor from different angles.

- **Matching** – Comparing one FMD against another FMD (usually the one created during the registration process) to see if they both represent the same fingerprint is called matching.

- **Identification** – Comparing a fingerprint FMD against a database of many stored fingerprint FMDs (typically, the fingerprints of all users of your application) to see if one or more of them matches is called identification. This technique allows your application to determine who is using it without having to request other forms of ID such as usernames or ID numbers.

- **Verification –** Using a fingerprint to confirm that a user is who they claim to be according to some other form of ID (such as a username or ID number) is called verification. Unlike other mechanisms such as passwords, swipe cards or PINs, fingerprints can't be lost, forgotten or shared.

- **Authentication** – The act of confirming that somebody is who they claim to be is called authentication. It usually involves two steps: (1) identifying who they say they are; and (2) verifying that they really are that person. When a fingerprint is used to both identify and verify somebody in one step, it is often called "touch-and-go" authentication.

- **False Match Rate (FMR) –** This is a measure of the probability that fingerprints from two different people might mistakenly be considered a match. A lower False Match Rate requires a more exact match, which could force legitimate users to rescan their fingerprints on occasion. Most applications allow this rate to be adjusted to handle different populations of users.

- **False Reject Rate (FRR) –** This is a measure of the probability that fingerprints from a legitimate user might mistakenly be rejected as not matching the ones previously enrolled, forcing the user to rescan. Typically, a lower False Match Rate will result in a higher False Reject Rate.

- **Failure To Capture (FTC) –** This occurs whenever a user presses their finger to the sensor and the sensor does not recognize that a finger is present. This can sometimes happen when people have very dry skin.

- **Duplicate Enrollment Check (DEC) –** This is the process of identifying individuals who have already registered their fingerprint with your application. This can be used during enrollment to make sure the user isn't being entered a second time.

- **User ID** – The piece of data that your application uses internally to identify each distinct user of your application is frequently called a "user ID." This unique identifier (often a form of user name or serial number) is used to quickly look up information about each person in whatever data store is used to record user information.

- **User Account Data** – Your application most likely stores information associated with each User ID in some sort of user account database. Typically, this includes attributes like account names, login names or IDs, ID numbers, PINs and other kinds of information that are used during sign-on.

Steps for Using Fingerprints in Your Application

To get the most out of fingerprint biometrics in your applications, focus on the following areas:

- Where to Store Fingerprint FMDs
- Accessing Stored Fingerprint FMDs
- Enrolling Users' Fingerprints
- Checking for Duplicate Enrollments
- Preloading FMDs at Application Startup
- Looking Up Users by Their Fingerprint
- Sign-on
- Fingerprints as an "Enter" key
- Approvals
- Sign-out
- Removing Users
- Logging

Fingerprints can be used to implement various security processes to make your application easier to use and more secure:

- **Identify users by their fingerprint** – Give users "touch-and-go" authentication without the need for other forms of ID, such as usernames, swipe cards or ID numbers.

- **Verify another form of ID** – Fingerprints can be used to confirm that a username or ID number provided by the user actually belongs to them. This avoids the need for passwords or PINs which can be easily lost, stolen or shared.

Most applications give customers' administrators the ability to set policies that control how users log onto the application. Common examples of logon policies include:

- Fingerprint-only
- Fingerprint or UserID+Password/PIN
- Fingerprint and UserID+Password/PIN

Your application implements the logic for these policies, giving you the flexibility to choose the most appropriate options for your customers.

Where to Store Fingerprint FMDs

The fingerprint FMDs that are created whenever a user enrolls fingerprints need to be stored in a way that your application can access them and know the user accounts to which they correspond.

Your existing user account data probably already has some form of User ID that can be used to quickly look up information about the user (e.g., a username or ID number). Fingerprints can provide a quick way to determine this User ID without having to ask the user for another form of ID.

There are two common approaches to choosing where fingerprint FMD data is placed:

Where	Extend Existing User Account Data	Use A Separate Database
How	Add fingerprint FMDs (at least two) as extra fields in the data you store about each UserID.	Store fingerprint FMDs in a separate database along with the UserID to which they correspond.
Pros	Takes advantage of your existing data backup and management tools.	Insulates fingerprint FMDs from user data for enhanced privacy and security.
Cons	Requires changes to existing user data structures.	Adds another database to backup and maintain.

Fingerprint FMDs are typically represented as binary data stored in variable-length arrays of bytes. The FMD format used most commonly with DigitalPersona software development kits is less than 2048 bytes long; however, other FMD formats have different sizes. If you are using Microsoft SQL Server, you can use a “`varbinary(3000)`” field.

Accessing Stored Fingerprint FMDs

If your application can be used by multiple people at the same time (such as from different computers, POS stations or other devices), you can minimize memory consumption and code complexity by creating a separate service for storing and looking up FMDs.

This service, which can even run on a separate computer, can be called by other parts of your application using technologies such as RPC, DCOM, WCF or Web Services. It provides an internal interface for your application to look up the User ID associated with a given fingerprint FMD. Keeping stored fingerprint data insulated from your end users also helps to protect people’s privacy.

Enrolling Users’ Fingerprints

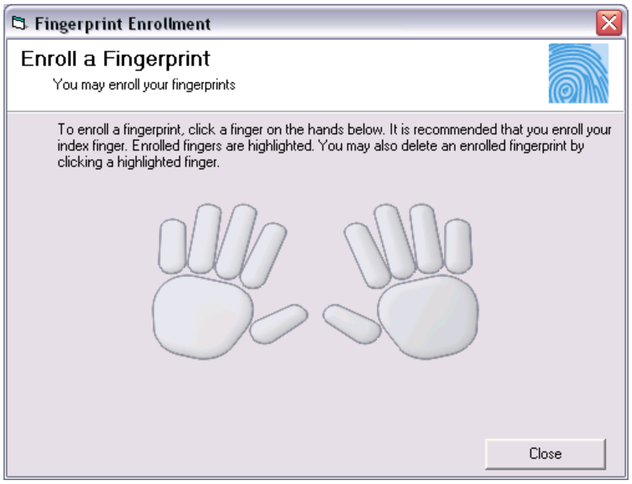
Each person who will be using fingerprints with your application has to enroll their fingerprints with your software. Many applications make this part of the user account creation or provisioning process. Typically, an administrator or other authorized user brings up the appropriate screen within your application and helps the user through their initial fingerprint scans.

The middle finger, index finger and thumb on each hand typically provide the best fingerprints to use.

To avoid matching problems in case of an injured finger, your application should ask users to register fingerprints from at least two fingers.

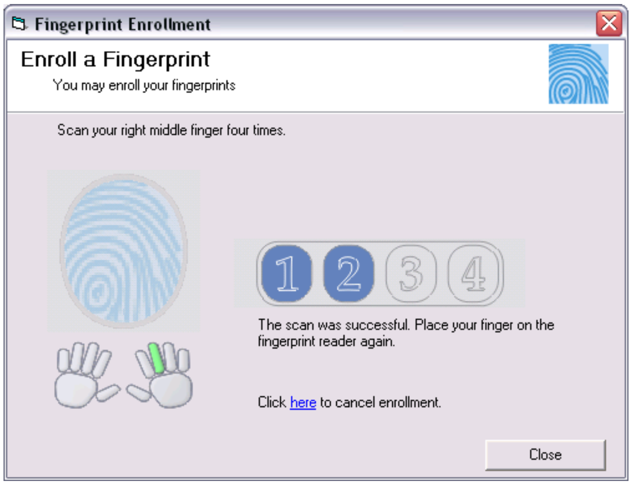
Graphical screens should be used to guide the user through the enrollment process. While touching a fingerprint reader is a natural, easy to understand action, including a picture or short video of somebody touching the pad (not the tip) of their finger to the surface of the fingerprint reader can avoid problems later.

DigitalPersona’s U.are.U Windows SDK includes graphical user interface controls that can either be used as is or to provide ideas if you wish to create your own interface:



To enroll a fingerprint, click a finger on the hands below. It is recommended that you enroll your index finger. Enrolled fingers are highlighted. You may also delete an enrolled fingerprint by clicking a highlighted finger.

If you create your own enrollment screens, make sure that users scan their fingerprints several times to make subsequent matches more reliable.



Scan your right middle finger four times.

The scan was successful. Place your finger on the fingerprint reader again.

Click here to cancel enrollment.

Also, make sure that your application provides visual feedback to the user.

Checking for Duplicate Enrollments

With biometrics, you can easily catch people who attempt to enroll more than once to use your application. This gives you the ability to consolidate older accounts, avoid accidental duplicate registrations and prevent fraudulent attempts to masquerade as somebody else.

Your application can either check for duplicates in real time during enrollment or offline (Footnote: FMDs of type DP_REGISTRATION cannot be cleansed offline. FMDs of ANSI and ISO can.) as part of a database cleansing process. Either approach can be implemented using the 1:n identification capability in the U.are.U SDKs and is useful even if your application only uses fingerprints to verify another form of ID.

Preloading FMDs at Application Startup

In order to ensure the fastest identification speeds\response times, do not read FMD data from a database at the point a user attempts to authenticate. You should pre-populate a list of FMDs at application startup by reading all FMD data from the database. When a user attempts to authenticate, all the application should do is call a single function – “identify()”.

When your application starts, have it iterate over the enrolled fingerprint FMDs (wherever you have chosen to store them) and add each one to a list, “identification collection” object. Once this is done, individual lookups will typically take less than a second, even when there are thousands of enrolled FMDs.

TIP If you are not using fingerprints for identification, but only to verify another form of ID, you do not need to use a 1:n identification.

Looking Up Users by Their Fingerprint

Fingerprints provide a natural way for your application to recognize the user without the need for other forms of ID (e.g., usernames, ID numbers or swipe cards). People learn quickly how to use fingerprints and can do so naturally, without having to stop or interrupt the flow of what they are doing. This makes fingerprints ideal not only for sign-on, but also for confirming who is performing important operations – especially when multiple people might be involved (such as for an approval).

Identification in U.are.U SDKs is not easily scalable as a service and is ideal for local identification as there is no memory or performance penalty during preload of FMDs. Under most circumstances, identification will be faster if the local machine performs the lookup as opposed to a server.

Your service or module may receive more than one possible match candidate.[1] By default, the returned list of candidates will be pre-sorted so that the first item in the list is likely the best match. Your application should log which users were matched and alert the administrator in the condition multiple candidates are returned during identification. Part of this messaging should include instructions on how to eliminate multiple matches, either by having the matched users re-register their finger(s) or by adjusting the False Match Rate.

TIP Once the appropriate enrolled FMD has been identified, your service or module can then return the UserID associated with the FMD to your application. You may wish to also return the enrolled FMD that

[1] Under certain conditions, a fingerprint FMD may partially match multiple enrolled FMDs, particularly if your application has lowered the False Match Rate to allow people with hard-to-read fingerprints to use your application without having to touch the fingerprint scanner multiple times.

was matched so that the caller can cache it for quick matching in the future.

Never implement fingerprint identification by iterating over your database of enrolled fingerprint FMDs, matching each one individually. This approach is very inefficient and will make users think your application is slow. Instead, utilize the Identification feature of the SDK to match the user in one operation.

Sign-On

The most common use of fingerprints is for sign-on, either as a form of identification or as a way to confirm another form of ID.

When a user scans their fingerprint during sign-on, your application will receive an event from the fingerprint SDK indicating that an image or FMD (depending on which SDK you are using) is available. If your application is using an SDK that provides a raw image, immediately extract the fingerprint FMD and discard the original image.

If you are using fingerprints as a form of ID, your sign-on code can call your lookup service or module (see above) to determine the UserID of the person who touched the fingerprint reader.

If you are only using fingerprints for verification, then your application can use the other form of ID to determine which UserID to look up. That UserID can then be used to find the user's enrolled FMDs to compare against.

If you will be using fingerprints to confirm actions that are performed frequently, obtain a copy of the enrolled fingerprint from your fingerprint lookup service or module and cache it in your application. Your code can then rapidly perform a direct match against the fingerprint in cache before attempting a full lookup.

Finally, always create a log entry whenever users sign on and note whether or not they used their fingerprint. Even if you do not create a policy requiring the use of fingerprints to sign on, it is still a good idea to note when anyone with registered fingerprints signs on without using them. This can help customers spot potential problems early.

Two-Finger Matching

For extra high security, you can request and match two fingerprints instead of just one. To avoid surprising users, always ask for both fingerprints, even if the first one correctly matches.

Fingerprints as an "Enter" Key

Fingerprints are useful for more than just sign-on. They are a fast, intuitive way for users to confirm that they are who they say they are when performing individual application functions, such as:

- Entering new orders
- Changing or deleting important data
- Opening a cash drawer in a cash register
- Printing sensitive information
- Accessing client credit card numbers

When you have an action that you want to confirm by a fingerprint, prompt the user to touch the fingerprint sensor and obtain a FMD as described above.

TIP Since most people tend to use the same finger over and over, if your application has previously cached the enrolled FMD that was successfully matched at sign on, try to match that cached FMD first.

If your application isn't caching any recently-used FMDs, or the FMD didn't match, do a look up using the approach described above for sign-on. This will tell you whether the fingerprint came from a different finger on the same person or from a different person.

If the fingerprint doesn't come from the user who signed on, you can use the FMD to determine if another authorized user is attempting to use your application. This is an easy way to implement approvals by supervisors or other privileged users (see next section).

Make sure that your application logs the fact that the action was confirmed with a fingerprint.

STOP As stated before, never iterate over all enrolled fingerprints looking for a match. It will make your application very slow, particularly as the number of users rises. Instead, use the 1:n identification capabilities of the U.are.U SDK to deliver a vastly superior user experience.

Approvals

Fingerprints can help guide users to follow proper business processes. They provide a simple way to allow other people (such as supervisors or administrators) to authorize actions requiring special permissions without cumbersome switching of users.

Your application implements the logic for approvals, giving you full control. For operations that require authentication from somebody with special privileges, provide a visual prompt explicitly identifying the privilege level required or the role of the person needed (e.g., "Manager Fingerprint Required for Override").

A simple way to implement approvals is to use the 1:n identification capabilities of the U.are.U SDK to identify which user scanned their fingerprint and, if that user is properly authorized, take the appropriate action. This eliminates the need to prompt for another form of identification (e.g., a username, login name, ID number or PIN) to determine which user has scanned a fingerprint. Workflow is fast and efficient and a powerful audit trail can be created.

STOP If you are not using the 1:n identification capabilities of the U.are.U SDK and don't wish to prompt for another form of ID, you will likely need to implement some form of persistent caching to avoid having to iterate over the list of all registered fingerprints. However, this adds significant complexity to your application and can greatly reduce performance.

Always have your application log all approval attempts – successful and failed.

Sign-Out

When a user signs out of your application, all temporary copies of fingerprint FMDs that your application is keeping in memory should be released. If your application is not using the 1:n identification capabilities of the U.are.U SDK but is maintaining its own persistent cache of registered fingerprint FMDs that have recently been used, make sure the cache is properly updated.

As always, make sure your application logs the fact that the user has signed out.

Removing Users

Using fingerprints to control access to your application makes it easy to immediately block access by people whose permissions have been revoked (*e.g.,* former employees or people who changed roles).

The easiest approach is to delete any FMDs associated with the former user. If your application uses the 1:n identification capabilities of the U.are.U SDK, remove the user from the identification collection that was created at startup to immediately prevent their fingerprints from being recognized. Then delete the registered FMDs from the user data record or from the separate fingerprint FMD database.

TIP If you wish to provide the ability for customers to flag terminated users who attempt to use their fingerprints to gain inappropriate access, do not immediately remove the user's fingerprint FMDs. Instead, mark the user's account data as disabled. Then, when a user attempts to access your application, simply check the status of that user's account to determine their access rights and log any failures.

STOP If your application does provide such temporary retention of biometric data, make sure that you give customers the ability to permanently flush the registered fingerprint FMDs from former users after a given number of days. Administrators should be able control the length of time and to immediately delete FMDs if needed. This is important as it enables the customer to comply with any local data retention regulations and policies.

Logging

Fingerprints are valuable as a deterrent to inappropriate behavior, as a way of improving usability and as a valuable source of data for an audit trail. Your application should automatically log all authentication and security activities, including:

- Whenever a user enrolls a fingerprint.
- Whenever somebody has trouble enrolling.
- Whenever a duplicate enrollment is detected.
- Whenever somebody signs on, confirms an action or otherwise authenticates – with the ways in which they authenticated.
- Whenever somebody tries to authenticate but can't.
- Whenever somebody who has fingerprints enrolled authenticates without them.
- Whenever security settings are changed – especially False Match Rate. Setting this improperly can have serious consequences. Make sure to include both the old and new values.

Troubleshooting and Preventing Problems

The following capabilities can simplify your customers' use of fingerprints and avoid common problems.

Provide visual feedback during fingerprint use

While fingerprints are naturally easy for people to understand, applications should provide feedback during successes as well as failures:

- Prompt the user when a fingerprint is needed.
- Warn when the sensor is disconnected.
- Prompt the user to retry if a finger is detected but no match is received within a second or two.
- Warn when a fingerprint is received but no match is found.
- Indicate success when a match is found.

Offer help when repeated failures occur

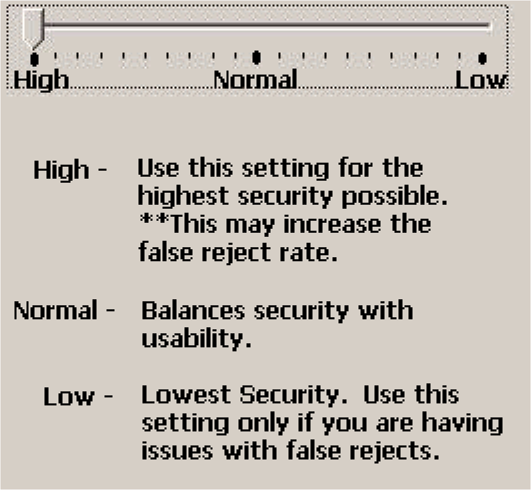
You can dramatically improve the user experience of your application by detecting repeated failed attempts to use the fingerprint reader and offering hints, such as:

- "Touch the fingerprint sensor with the flat pad of your finger, not the tip."
- "If your fingers are very dry, try touching your forehead with the pad of the finger you are trying to scan and then rescanning your fingerprint."
- "If the fingerprint sensor is dirty, gently dab it with the sticky side of a piece of cellophane tape. Do not rub it with paper and do not get it wet."

Provide administrative settings for FMR

For some populations of users, the default False Match Rate settings might be too restrictive or too forgiving, particularly when fingerprints are used for identification (for verification, the settings rarely need to be changed).

If you are using fingerprints for identification, you should provide a way for administrators (but not end users) to adjust the FMR settings. For example:



High - Use this setting for the highest security possible. **This may increase the false reject rate.

Normal - Balances security with usability.

Low - Lowest Security. Use this setting only if you are having issues with false rejects.

Your application can map High or Low settings to the appropriate values needed by the appropriate SDKs.

STOP Incorrectly adjusting the FMR can have serious consequences. It is extremely important that your application set the FMR according to the SDK documentation and provide administrators a thorough explanation of FMR options within your user interface to avoid confusion.

Test Your Application with Multiple People

The ease with which people's fingerprints can be read is affected by many factors, including dryness, age, as well as wear and tear. For best results, try your implementation with multiple and diverse people.

Summary

Biometrics can help you enhance the security and usability of your application. By following a few simple guidelines and using DigitalPersona's biometric software development kits, you can easily add fast fingerprint identification and verification capabilities that enable your application to recognize individual users without requiring other forms of ID. This can be used in a variety of ways – from sign-on and confirmation of important actions to special approvals by other users – to help combat fraud and boost customer efficiency.

About DigitalPersona, Inc.

DigitalPersona, Inc. provides biometric solutions which securely create, manage and authenticate the digital identities of individuals and businesses worldwide. The company's fingerprint biometrics technology helps organizations prevent fraud and increase accountability; it is incorporated into multiple national voting systems, almost all brands of biometrically-enabled point-of-sale (POS) stations, as well as many commercial applications in the retail, healthcare and financial industries. DigitalPersona's authentication and access management software is shipped by computer manufacturers on millions of notebooks and desktop computers per year; its cloud- and Active Directory-managed solutions multi-factor/strong authentication, single sign-on (SSO) password management and emergency access recovery simplify compliance and cut IT costs.

For more information contact DigitalPersona, Inc. at: +1 650.474.4000, or visit www.digitalpersona.com

The information in this document is provided for informational purposes only and is not intended to be anything other than the educated opinion of the author. All statements, information and recommendations in this document are believed to be accurate, but DigitalPersona makes no claims, promises or guarantees about the accuracy, completeness or adequacy of the information. This information should not be relied upon as legal advice. DigitalPersona specifically disclaims all warranties of any kind, express or implied. Users must take full responsibility for their application or any products or compliance with any legal requirements.