



WF Fingerprint reader User's Manual

AC0061-004 rev E



Contents

Support information	2
Technical support and Product information	2
WireFlow headquarters.....	2
Introduction	3
How it works	4
How to place your finger on the sensor.....	5
Quick tips.....	5
Finger angle (pitch)	5
Finger roll	6
Translation	6
Rotation	7
The WF Fingerprint driver	8
Requirements	8
Installation	8
Device driver for Windows.....	8
Installed items	9
Capture image.....	9
Database.....	10
Enrol	11
Check	11
Error Codes	12
Deploying applications.....	13
.inf file	13
Troubleshooting	14
When running an application, I get Error 6369	14
1. Verify that windows can find the WF2111 (device driver is installed).....	14
2. Verify that NI-VISA is installed.....	14
3. Verify that VISA USB Runtime Support is installed and USB Passport is enabled. 14	14

Support information

Technical support and Product information

www.wireflow.se/products/wf2111

support@wireflow.se

WireFlow headquarters

WireFlow AB
Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

© WireFlow AB, 2017

Introduction

The WireFlow Fingerprint reader adds a convenient biometric login and user identification to your LabVIEW applications. It includes all hardware and software needed to get started. It is also compatible with TestStand user management.

The WireFlow Fingerprint reader uses a high quality robust fingerprint sensor with a thick, hard and scratch resistant coating. The sensor is mounted in an ergonomic sensor housing that includes electronics for sending fingerprint images to the host via the USB cable.



The included LabVIEW driver provide VI:s for USB communication and state-of-the-art biometric algorithms for enrolment and identification.

If you are to create a LabVIEW application that will limit or enable application features based on who is currently logged in, we recommend you to also use the *WF User Access Toolkit* together with the fingerprint reader. This toolkit includes intuitive pre-programmed dialogs to manage users, groups and assign permissions.

How it works

The WF Fingerprint reader uses a CMOS fingerprint sensor, which delivers superior image quality with 256 gray scale values in every single programmable pixel. The reflective measurement method sends an electrical signal via the frame directly into the finger. This technique enables the use of an unbeatable hard and thick protective surface coating. The sensor with its 3D pixel sensing technology can read virtually any finger; dry or wet.

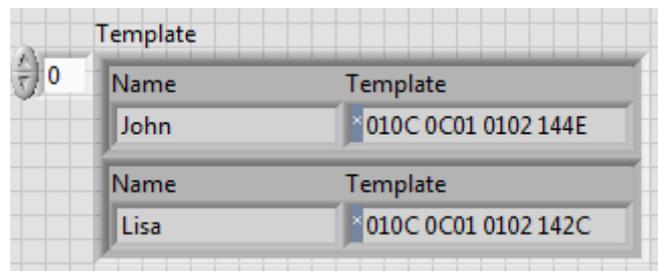
Thanks to the extremely hard and durable surface coating, the reader is capable of more than 10 million finger placements. The reader is also protected against ESD well above 30 kV as well as scratches and everyday wear-and-tear.

The ergonomic housing and the guidance frame, simplifies proper fingerprint guidance and hence improving algorithm performance.

By using the WF Fingerprint LabVIEW driver you can capture an image with the *Capture Image.vi* on the host computer. When doing this the sensor takes an image of the finger and transfers it through the USB cable in to the host computer.

With the *Check.vi* you can check if the finger exists in the fingerprint database.

The fingerprint database is managed by the LabVIEW driver. It stores so called fingerprint templates, which describes the significant characteristics of fingerprints. The database contains a name- and a template string for each fingerprint. The database array looks like this:

A screenshot of a LabVIEW array structure. The array is titled "Template" and contains two rows of data. Each row has two columns: "Name" and "Template". The first row contains "John" and "010C 0C01 0102 144E". The second row contains "Lisa" and "010C 0C01 0102 142C".

Template	
Name	Template
John	010C 0C01 0102 144E
Lisa	010C 0C01 0102 142C

Finally the LabVIEW driver also provides a VI that creates the fingerprint templates from the fingerprint images. This VI is called *Enrol.vi*.

How to place your finger on the sensor

This section describes how to place the finger on the sensor to get an accurate, high quality image of the fingerprint. It also points out the most common user difficulties when performing enrolment and verification.

Quick tips

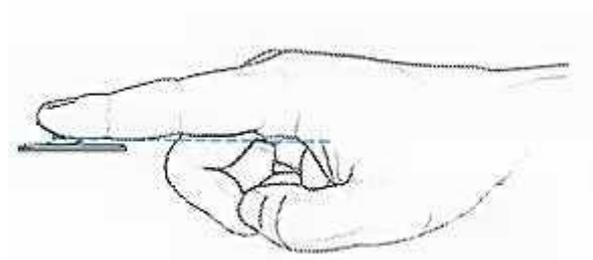
To improve enrolment and identification performance, the following factors should be considered each time the finger is placed on the sensor:

1. Repeatability: The finger should preferably be placed in the same position every time.
2. The image should contain as much of the fingerprint core area as possible.
3. The pressure has to be high enough to create a satisfying contact interface between the finger and the sensor surface.

Finger angle (pitch)

Pitch occurs when only the upper part of the fingertip is placed on the sensor, as if the user points at something.

When placing the finger with a high angle, only the tip is captured. The fingertip does not contain enough information for verification. It is very important that the angle between the finger and the sensor is as small as possible.



Correct finger angle



Incorrect finger angle

Finger roll

Finger roll is defined as an incorrect placement regarding rolling the finger sideways. As the situation above (finger pitch), the fingerprint recognition system needs to receive enough information, preferably as much of the fingerprint core area as possible. If the finger is rolled, less useful information is received, degrading the matching performance.

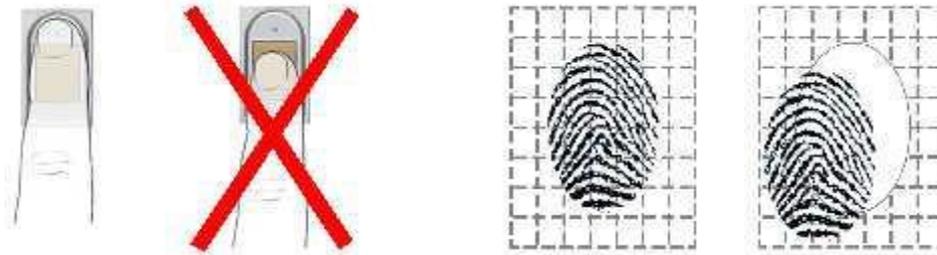


Incorrect finger roll

Translation

Translation can be defined as the difference in horizontal and vertical distance from the enrolled template to the verification occasion.

The algorithm can handle some translation, but large translations should be avoided for best performance.

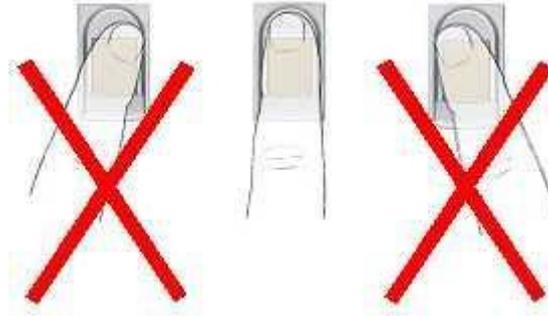


Example of good and bad finger placement

Rotation

Rotation is defined as the angle difference between an imagined centreline of the finger from one occasion to another.

The algorithm allows some variation from the enrolment occasion, but in general, rotation should be avoided as much as possible.



Example of good and bad finger rotation

The WF Fingerprint driver

This easy-to-use LabVIEW driver is used to manage fingerprints in your LabVIEW applications.

Requirements

WireFlow USB fingerprint reader WF 2111

LabVIEW (version \geq 2012)

NI-VISA (version \geq 5.4)

NI-VISA Runtime Support for USB (select this when installing NI-VISA)

VI Package Manager (for installation)

Currently the LabVIEW driver is only compatible with LabVIEW (both 32- and 64-bit) for Windows.

Installation

The WF Fingerprint driver may be downloaded from www.wireflow.se. For the installation you will need the VI Package Manager which may be downloaded from www.ni.com or from jki.net. The installation procedure is quite straightforward. Just follow the online instructions.

Device driver for Windows

To be able to run the code against the USB interface on Windows platform you also need to install the WF Fingerprint reader driver for Windows.

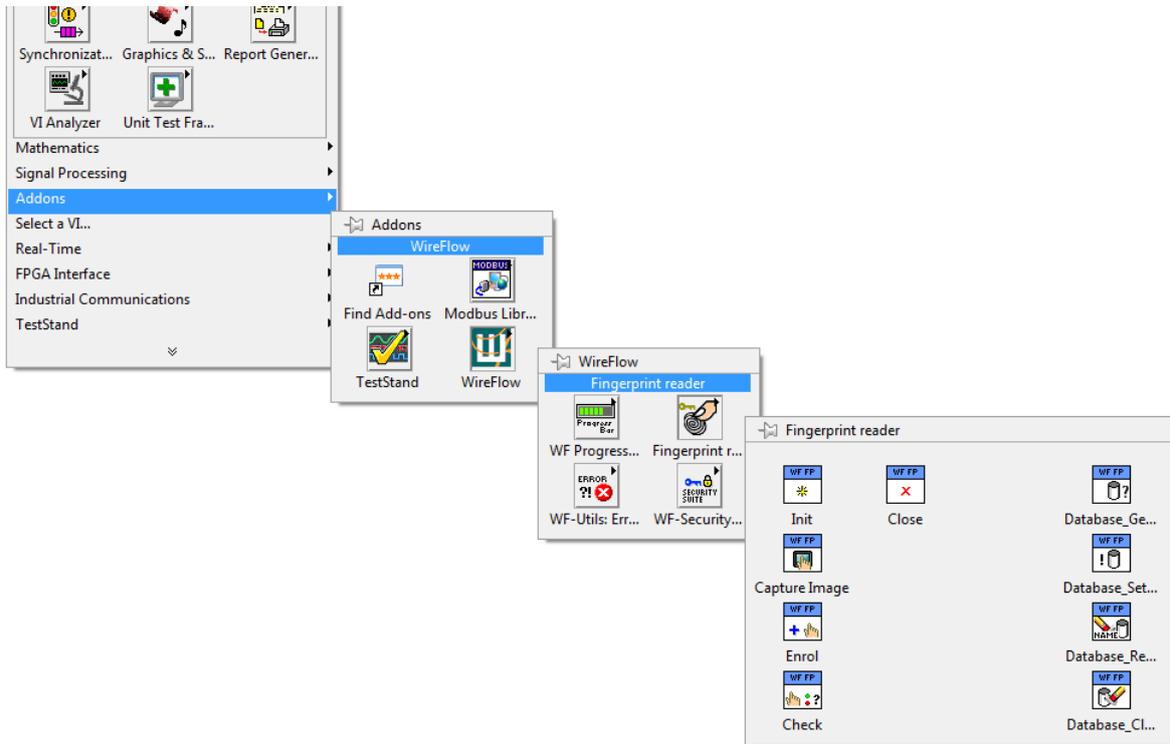
The device driver is an .inf file that can be found on your computer in the folder:

```
C:\ProgramData\WireFlow\Fingerprint reader\Device driver\Windows Device driver
```

The device driver .inf file can also be downloaded from www.wireflow.se. Follow the installation instruction located together with the .inf file.

Installed items

The driver VIs are found in the functions palette:



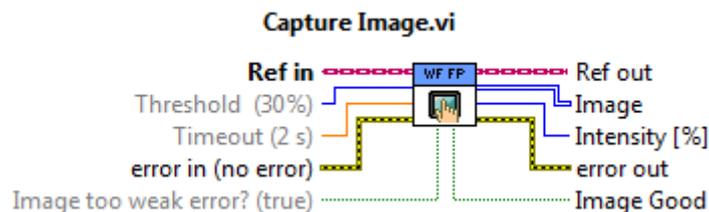
The user's manual is installed and found under the menu >Help >WireFlow

Examples are found under the menu >Help > Find Examples...

To find the examples select the Search tab and then perform a search for the keywords “fingerprint” or “WireFlow”

Capture image

When you execute the *Capture Image.vi* on the host computer the sensor takes an image of the finger and then transfers the image through the USB cable in to the host computer. The *Capture Image.vi* returns the image and an indication of the image intensity, 0-100%. It is recommended that an image shall have an intensity above 30% to be used for further analysis. The reason for an image to have low intensity is normally that the finger is pressed to lose against the sensor.



Database

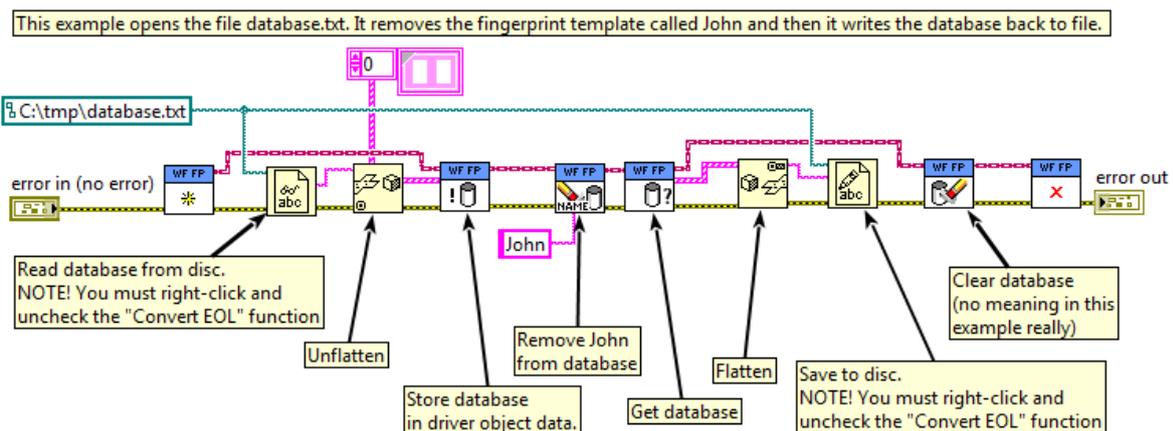
You will use a database of fingerprints to check who the finger on the sensor belongs to. The database does not store the actual fingerprint image, but instead it stores a template for the fingerprint. This template is a description of the significant characteristics of a specific fingerprint. The database contains a name string and a template string for each fingerprint. The database is actually an array kept in the WF Fingerprint LabVIEW driver. The array looks like this:

Template	
Name	Template
John	010C 0C01 0102 144E
Name	Template
Lisa	010C 0C01 0102 142C

The WF fingerprint driver provides four VI:s to manage the database;

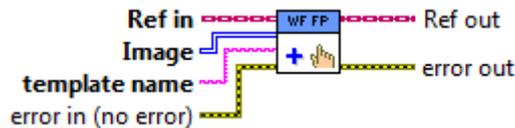
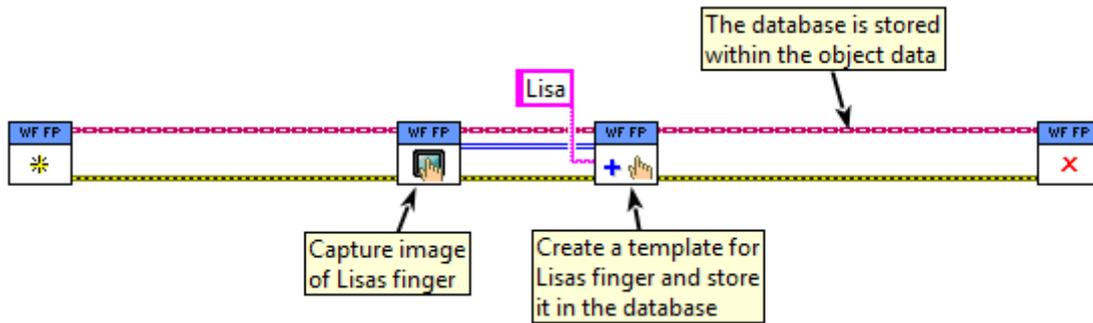
- *Database_GetTemplates.vi*
- *Database_SetTemplates.vi*
- *Database_RemoveTemplateByName.vi*
- *Database_clear.vi*.

By using these database VI:s together with the LabVIEW functions for Flatten/Unflatten strings and Write/Read Text files you can easily store your database on disc. The following diagram illustrates how this can be done.



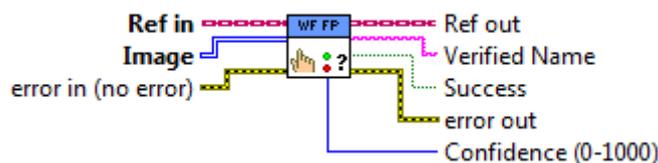
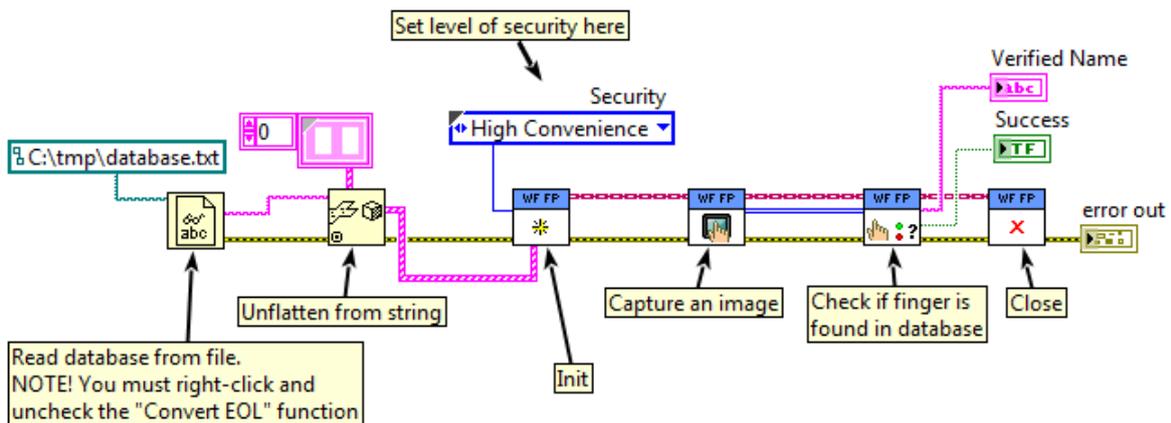
Enrol

If you want to add a new fingerprint to the database you must first take an image with *Capture Image.vi* and then wire the image to the input on *Enrol.vi*. You must also specify the desired name for the new template. The *Enrol.vi* will now analyse the image and create a template for the finger by identifying the unique characteristics found in the fingerprint image. When this analyse is done the *Enrol.vi* will add the new template to the database that is held in the object data.



Check

To check if a captured fingerprint exists in the database you need to execute the *Check.vi*. First take an image with *Capture Image.vi* and then wire the image to the input on *Check.vi*. The *Check.vi* will now search the database for a match. How certain it must be for setting a match is specified with the Security input on the *Init.vi*. The best match found will be returned if it fulfils the security level. The confidence of the match will also be returned.



Error Codes

The WireFlow fingerprint driver VIs can return one of the following custom error codes

Code	Source text	Explanation
6391	Image too weak!	The image contrast is lower than the requested value.
6392	Template name cannot be empty!	Adding a template without specifying a name is not possible.
6393	No supported fingerprint device found!	No valid fingerprint reader device found in the system, please check that the device is present in MAX.
6394	Database is empty!	No checks can be done if the template database is empty.
6395	Illegal Image Format!	The size of the 2D array is not recognized by the algorithm.
6396	Enrolment error; please check the fingerprint image data. (Internal error %d)	The contrast in the image data is probably too low. Get a new fingerprint image and use a higher threshold value.
6397	GetTemplateData error; please check the template data string. (Internal error %d)	The template string is invalid. Only template data generated by the WF fingerprint driver can be used.
6398	Match error; please check template and image data. (Internal error %d)	Image data and/or template data is invalid. Check the contrast of the input image, and note that only templates generated with WF fingerprint driver can be used.
6399	SetImageBuffer error; please check that the image is valid. (Internal error %d)	The input image is invalid, e.g. an empty string or of an unknown format.

Deploying applications

This chapter gives some guidelines for how to deploy LabVIEW applications that uses the WF fingerprint reader.

.inf file

If you are creating a Windows application then you need to install the .inf file on the target computer.

1. Create an installer for your application which will install an .exe file on the target computer.
2. Since your application is to be run on a Windows target the WF Fingerprint reader device driver for Windows (the .inf file) must also be installed on the target computer. If your end customer is supposed to do the installation himself then you must provide the .inf file to the end user so that he can install it on his machine.

Troubleshooting

When running an application, I get Error 6393

This can have multiple causes, most commonly though is that NI-VISA isn't properly installed or configured. Follow this guide to verify that it is.

1. Verify that windows can find the WF2111 (device driver is installed)

From start menu, search for and open "Device Manager".

- If WF2111 is found under "NI-VISA USB Devices", the **device driver is installed**, → OK



- If WF2111 is found under "Other devices" the **device driver is NOT installed**, → see chapter "Installation" for installation instructions.

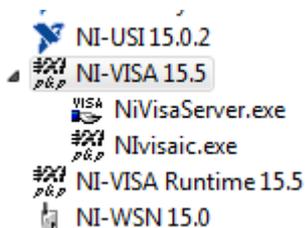


- If WF2111 is not present at all, Windows can't find it. → unplug USB and plug back in.

2. Verify that NI-VISA is installed

Open MAX and expand "My System/Software"

- If "NI-VISA" and "NI-VISA Runtime" are listed, **NI-VISA is installed**, → OK



- If either of them is missing, **NI-VISA is NOT installed** completely, →install NI-VISA (See chapter Rrequirements) from your NI installation media.

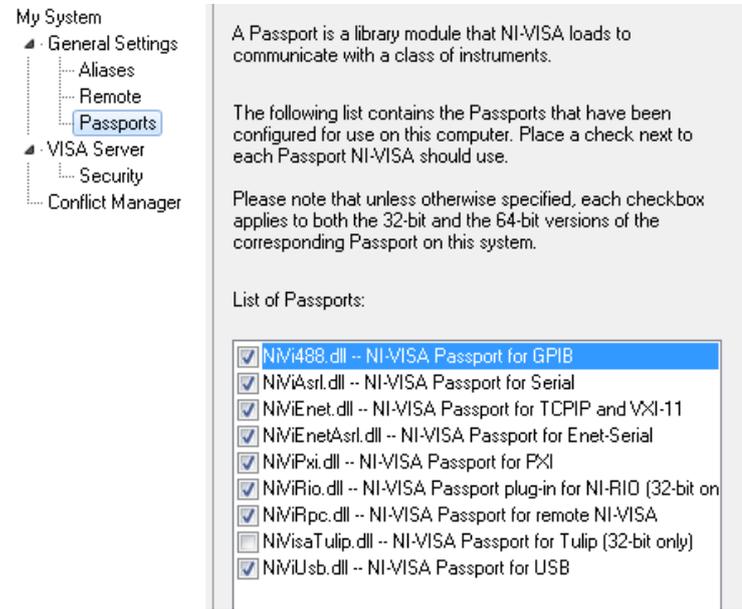
3. Verify that VISA USB Runtime Support is installed and USB Passport is enabled

Open MAX, expand "My System/Software" and select "NI-VISA".

In the right pane, select “VISA Options”.

In the right pane, expand “MySystem/General Settings/Passports”.

- If NiViUsb.dll is listed and checked, **USB Runtime Support is installed**,
→ OK



- If NiViUsb.dll is missing, **USB Runtime Support is NOT installed**,
→ install NI-VISA Runtime Support for USB. Use your NI installation media or from “Control Panel” in Windows.
- If NiViUsb.dll is unchecked,
→ check it