How to use CardSwipe II

CardSwipe II® is an iOS app that bridges the gap between Magstripe readers, EMV Chip readers, and your software. You can use CardSwipe II® to read credit cards, debit cards, gift cards, loyalty cards, driver's licenses, ID badges, and more.

CardSwipe II® is secure and safe. It functions strictly as a pass through. No card data is stored in CardSwipe II®.

New Features

VivoPay 3300 • 3-in-1 MagStripe, EMV Contact and EMV Contactless Reader

NEW – Versions…

IDMR-BT93133AP - VP3300 Bluetooth - AES encryption - Black

IDMR-BT93133APW - VP3300 Bluetooth AES encryption - White

They can be ordered from: ID Tech IDMR-BT93133AP - Barcodesinc.com

Branding – You can replace CardSwipe logo and Company name with your Logo and Business name.

Deprecated

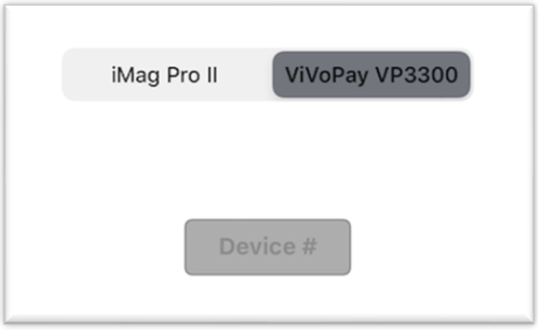
UniPay 1.5 • CardSwipe II® no longer supports the IDTech reader. The latest line of iOS devices no longer equipped audio jacks. The reader was also known to work inconsistently.

iMag Pro II • Lightning Connector iOS Magstripe Reader – discontinued.

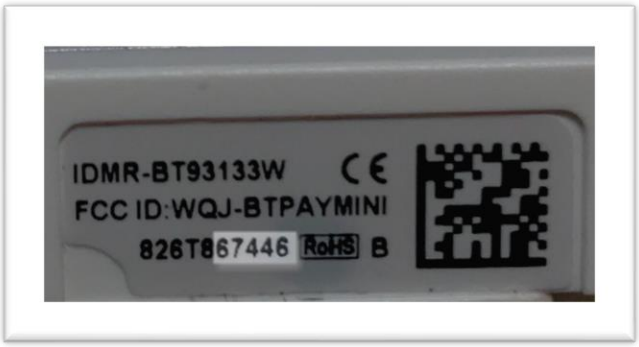
Connecting VivoPay VP3300

Each VivoPay device needs to be registered with CardSwipe II:

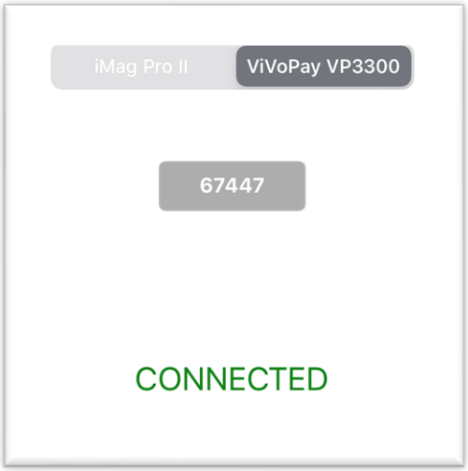
1. Launch CardSwipe II
2. Tap on VivoPay VP3300



3. Enter the last 5 digits of the serial #



4. Resulting in…



5. Insert the chip facing the IDTech logo on the VivoPay 3300

How it works

These instructions are based on script calls from your FileMaker Database using the Open URL [] function. Two scripts are required. One to call CardSwipe II. The second to receive the response from CSII.

CSII uses iOS Protocol - CSII = CSII://? • FileMaker = fmp://

(Note: II are Capital "i"s)

CardSwipe II has 2 read types:

Swipe/Dip - for magstripe and chip cards – **CSII://?SwipeDip?** - (Chip Read or Swipe)

Tap* - for Contactless payments - **CSII://?Tap? -** (Apple Pay, Samsung Pay, etc.)

* Only the VivoPay VP3300 supports Tap transactions. AMEX cards may work but are not supported.

Creating the URL Call

Open URL[] starts with one of the CSII read types. This call activates CSII for the requested read type.

Open URL [**CSII://?SwipeDip?**] or Open URL [**CSII://?Tap?**]

After the card is read we need to send that data back to your FileMaker DB. FM Protocol is appended to the read type referenced above.

CSII Read Type:

Open URL [**CSII://?SwipeDip?fmp://**

Database location: IP address of the hosting computer or a tilde ~ if the FM GO database is on the iOS device

Open URL [**CSII://?SwipeDip?**fmp://**192.168.1.1**/ **or**

Open URL [**CSII://?SwipeDip?**fmp://~/

File Name: File names are case sensitive. You do not need the file extension .fmp12

Open URL [**CSII://?SwipeDip?**fmp://192.168.1.1/**YourFileName?]**

Script Name: We named our script to run "PostData"

Open URL [**CSII://?SwipeDip?**fmp://192.168.1.1/YourFileName?**script=PostData]**

Results in FileMaker

The FileMaker script that receives the results will receive 4 variables: $CardData, $Track1, $Track2, and $Track3. These variables will be available as soon as the script runs, you can just call them, there's no need to declare them. If a variable is nonexistent, it's because there was no data on that card for that variable.

The data in each variable will be formatted as follows:

$CardData = 4444333322221111_LAST-FIRST-MIDDLEINITIAL_MMYY

Card Number Name Expiration Date

Parse $CardData – Substitute the underscore with a Return. Then GetValue() to parse each element.

```
Set Variable [ $SeparatedCardData ; Value: Substitute ( $CardData ; "_" ; ¶ )
Set Field [ Cards::CardNumber; GetValue ( $SeparatedCardData ; 1 ) ]
Set Field [ Cards::CardName; GetValue ( $SeparatedCardData ; 2 ) ]
Set Field [ Cards::CardExpMth; Left ( GetValue ( $SeparatedCardData ; 3 ) ; 2 ) ]
Set Field [ Cards::CardExpYr; Right ( GetValue ( $SeparatedCardData ; 3 ) ; 2 ) ]
```

Track data varies from card to card. There are characters used in the Tracks that cannot be use by the iOS URL Protocol. If they are required when capturing the sale, they can be added back in your FileMaker Script.

$Track1 = Track 1 data. "%" character at the beginning is omitted. The "^" characters are replaced by a "_"

$Track2 = Track 2 data. The "=" character is replaced by a ":"

$Track3 = Track 3 data does not have invalid characters.

FileMaker Script Example:

```
#Track variables
Set Field [ Cards::Track1; $Track1 ]
Set Field [ Cards::Track2; $Track2 ]
Set Field [ Cards::Track3; $Track3 ]
```

Using CardSwipe® with a Web Page

Using CardSwipe II to pass data to a web page requires a page that can receive the data and parse it and load it into your form fields.

The URL call would be constructed as outlined above except for the Database Location. This will be replaced with the path to your data processing page*.

https://www.yourwebsite.com/webpagetoprocessdata.php

https:// is required to meet PCI compliance when sending credit card data and is recommended if passing any sensitive data.

Append your return URL path to the web page that is processing the swipe followed by a data tag. The complete call to CardSwipe would be something like this.

CSII://?SwipeDip?https://www.yourwebsite.com/webpagetoprocessdata.php?

The processing web page receive the 4 variables specified above.

* Check our website https://ccq-fm.com for future updates and sample php code.

Branding - Load Company Logo and Company Name

CardSwipe II® can be branded with your logo and company name to display. Use the following Open URL[] call.

Open URL [**CSII://**?&userName=**companyName**&logo=**encodedLogo**]
(Replace “companyName” and “encodedLogo” with your data)

You must encode your logo image using base64 encoding. You can set up a variable or a calculation field to do this as follows: (replace **LogoField** with the container field that has your logo).

```
Let ( [
~encoded = Base64Encode ( LogoField ) ;
~sub10 = Substitute ( ~encoded ; Char ( 10 ); "" ) ;
~encodedLogo = Substitute ( ~sub10 ; Char ( 13 ); "" ) ]
; ~encodedLogo )
```

Please refer to your applications documentation for further information.