IDTech iOS SDK Guide

for UniPay

#80131513-001
Rev. A

Revision History

Revision	Description and Reason for Change	Date
A	Initial Release - Manual;User;UniPay;SDK;iOS	2/23/2015
A	Updated Permission Requirements	3/16/2017

Contents

Chapter 1

IDTech iOS Framework Reference Guide for UniPay

The IDTech Framework is an Apple Framework that will be provided by IDTech as the main interface between iOS applications, the UniPay and payment processing solutions.

The purpose of this document is to describe the requirements of the frameworks as well as the interface definitions and requirements needed for any iOS applications wishing to deploy with the payment application.

- EMV Tag Reference
- Enumeration Reference
- UniPay Error Code Reference

Chapter 2

Important Security Notice

The Payment Card Industry Payment Application Data Security Standard (PCI PA-DSS) is comprised of fourteen requirements that support the Payment Card Industry Data Security Standard (PCI DSS). The PCI Security Standards Council (PCI SSC), which was founded by the major card brands in June 2005, set these requirements in order to protect cardholder payment information. The standards set by the council are enforced by the payment card companies who established the Council: American Express, Discover Financial Services, JCB International, MasterCard Worldwide, and Visa, Inc.

PCI PA-DSS is an evolution of Visas Payment Application Best Practices (PABP), which was based on the Visa Cardholder Information Security Program (CISP). In addition to Visa CISP, PCI DSS combines American Express Data Security Operating Policy (DSOP), Discover Networks Information Security and Compliance (DISC), and MasterCards Site Data Protection (SDP) into a single comprehensive set of security standards. The transition to PCI PA-DSS was announced in April 2008. In early October 2008, PCI PA-DSS Version 1.2 was released to align with the PCI DSS Version 1.2, which was released on October 1, 2008. On January 1, 2011, PCI PA-DSS Version 2.0 was released. This extends the PCI DSS Version 1.2, which was released on October 1, 2008 and is effective as of January 1, 2011.

2.1 Applicability

The PCI PA-DSS applies to any payment application that stores, processes, or transmits cardholder data as part of authorization or settlement, unless the application would fall under the merchants PCI DSS validation. It is important to note that PA-DSS validated payment applications alone do not guarantee PCI DSS compliance for the merchant. The validated payment application must be implemented in a PCI DSS compliant environment. If your application runs on Windows XP, you are required to turn off Windows XP System Restore Points.

2.2 What Does PA-DSS Mean to You?

The following table provides opening points to cover in any discussion with merchants on data storage.

	Data Element	Storage Permitted	Protection Required	PCI DSS Req. 3, 4
Cardholder Data	Primary Account Number	Yes	Yes	Yes
	Cardholder Name [1]	Yes	Yes [1]	No
	Service Code [1]	Yes	Yes [1]	No
	Expiration Date [1]	Yes	Yes [1]	No
Sensitive Authentication Data [2]	Full Magnetic Stripe Data [3]	No	N/A	N/A
	CAV2/CID/CVC2/CVV2	No	N/A	N/A
	PIN/PIN Block	No	N/A	N/A

[1] These data elements must be protected if stored in conjunction with the PAN. This protection should be per PCI DSS requirements for general protection of the cardholder environment. Additionally, other legislation (for example, related to consumer personal data protection, privacy, identity theft, or data security) may require specific protection of this data, or proper disclosure of a company's practices if consumer-related personal data is being collected during the course of business. PCI DSS, however, does not apply if PANs are not stored, processed, or transmitted.

[2] Do not store sensitive authentication data after authorization (even if encrypted).

[3] Full track data from the magnetic stripe, magnetic-stripe image on the chip, or elsewhere.

2.3 Third Party Applications

The end-to-end transaction process, beginning with entry into the third party application until the response from the payment engine is returned, must meet the same level of compliance. In order to claim the third party application is end-to-end compliant, the application would need to be submitted to a QSA for a full PA-DSS audit.

The end user and/or P.O.S. developer can integrate and be compliant in the processing portion of a payment transaction. A brief review (given below) of the PA-DSS environmental variables that impact the end user merchant can help the end user merchant obtain and/or maintain PA-DSS compliance. Environmental variables that could prevent passing an audit include without limitation issues involving a secure network connection(s), end user setup location security, users, logging and assigned rights. Remove all testing configurations, samples, and data prior to going into production on your application.

2.4 PA-DSS Guidelines

The following PA-DSS Guidelines are being provided by IDTech as a convenience to its customers. Customers should not rely on these PA-DSS Guidelines, but should instead always refer to the most recent PCI DSS Program Guide published by PCI SSC.

1. Sensitive Data Storage Guidelines

Do not retain full magnetic stripe, card validation code or value (CAV2, CID, CVC2, CVV2), or PIN block data. 1.1 Do not store sensitive authentication data after authorization (even if encrypted): Sensitive authentication data includes the data as cited in the following Requirements 1.1.1 through 1.1.3. PCI Data Security Standard Requirement 3.2

Note: By prohibiting storage of sensitive authentication data after authorization, the assumption is that the transaction has completed the authorization process and the customer has received the final transaction approval. After authorization has completed, this sensitive authentication data cannot be stored.

1.1.1 After authorization, do not store the full contents of any track from the magnetic stripe (located on the back

of a card, contained in a chip, or elsewhere). This data is alternatively called full track, track, track 1, track 2, and magnetic-stripe data.

In the normal course of business, the following data elements from the magnetic stripe may need to be retained:

- The accountholders name,
- Primary account number (PAN),
- Expiration date, and
- Service code
- To minimize risk, store only those data elements needed for business.

Note: See PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms for additional information. PCI Data Security Standard Requirement 3.2.1

1.1.2 After authorization, do not store the card-validation value or code (three-digit or four-digit number printed on the front or back of a payment card) used to verify card-not-present transactions. Note: See PCI DSS and P↩ A-DSS Glossary of Terms, Abbreviations, and Acronyms for additional information. PCI Data Security Standard Requirement 3.2.2

1.1.3 After authorization, do not store the personal identification number (PIN) or the encrypted PIN block.

Note: See PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms for additional information. PCI Data Security Standard Requirement 3.2.3

1.1.4 Securely delete any magnetic stripe data, card validation values or codes, and PINs or PIN block data stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion, as defined, for example by the list of approved products maintained by the National Security Agency, or by other State or National standards or regulations. PCI Data Security Standard Requirement 3.2

Note: This requirement only applies if previous versions of the payment application stored sensitive authentication data.

1.1.5 Securely delete any sensitive authentication data (pre-authorization data) used for debugging or troubleshooting purposes from log files, debugging files, and other data sources received from customers, to ensure that magnetic stripe data, card validation codes or values, and PINs or PIN block data are not stored on software vendor systems. These data sources must be collected in limited amounts and only when necessary to resolve a problem, encrypted while stored, and deleted immediately after use. PCI Data Security Standard Requirement 3.2

2. Protect stored cardholder data

2.1 Software vendor must provide guidance to customers regarding purging of cardholder data after expiration of customer-defined retention period. PCI Data Security Standard Requirement 3.1

2.2 Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed).

Notes:

- This requirement does not apply to those employees and other parties with a legitimate business need to see full PAN;
- This requirement does not supersede stricter requirements in place for displays of cardholder datafor example, for point-of-sale (POS) receipts. PCI Data Security Standard Requirement 3.3

2.3 Render PAN, at a minimum, unreadable anywhere it is stored, (including data on portable digital media, backup media, and in logs) by using any of the following approaches:

- One-way hashes based on strong cryptography with associated key management processes and procedures
- Truncation

- Index tokens and pads (pads must be securely stored)
- Strong cryptography with associated key management processes and procedures. The MINIMUM account information that must be rendered unreadable is the PAN. PCI Data Security Standard Requirement 3.4

The PAN must be rendered unreadable anywhere it is stored, even outside the payment application. Note: Strong cryptography is defined in the PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms.

2.4 If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed independently of native operating system access control mechanisms (for example, by not using local user account databases). Decryption keys must not be tied to user accounts. PCI Data Security Standard Requirement 3.4.2

2.5 Payment application must protect cryptographic keys used for encryption of cardholder data against disclosure and misuse. PCI Data Security Standard Requirement 3.5

2.6 Payment application must implement key management processes and procedures for cryptographic keys used for encryption of cardholder data. PCI Data Security Standard Requirement 3.6

2.7 Securely delete any cryptographic key material or cryptogram stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion, as defined, for example the list of approved products maintained by the National Security Agency, or by other State or National standards or regulations. These are cryptographic keys used to encrypt or verify cardholder data. PCI Data Security Standard Requirement 3.6

Note: This requirement only applies if previous versions of the payment application used cryptographic key materials or cryptograms to encrypt cardholder data.

3. Provide secure authentication features

3.1 The payment application must support and enforce unique user IDs and secure authentication for all administrative access and for all access to cardholder data. Secure authentication must be enforced to all accounts, generated or managed by the application by the completion of installation and for subsequent changes after the "out of the box" installation (defined at PCI DSS Requirements 8.1, 8.2, and 8.5.88.5.15) for all administrative access and for all access to cardholder data. PCI Data Security Standard Requirements 8.1, 8.2, and 8.5.88.5.15

Note: These password controls are not intended to apply to employees who only have access to one card number at a time to facilitate a single transaction. These controls are applicable for access by employees with administrative capabilities, for access to servers with cardholder data, and for access controlled by the payment application. This requirement applies to the payment application and all associated tools used to view or access cardholder data. 3.1.10 If a payment application session has been idle for more than 15 minutes, the application requires the user to re-authenticate. PCI Data Security Standard Requirement 8.5.15.

3.2 Software vendors must provide guidance to customers that all access to PCs, servers, and databases with payment applications must require a unique user ID and secure authentication. PCI Data Security Standard Requirements 8.1 and 8.2

3.3 Render payment application passwords unreadable during transmission and storage, using strong cryptography based on approved standards

Note: Strong cryptography is defined in PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms. PCI Data Security Standard Requirement 8.4

4. Log payment application activity

4.1 At the completion of the installation process, the out of the box default installation of the payment application must log all user access (especially users with administrative privileges), and be able to link all activities to individual users. PCI Data Security Standard Requirement 10.1

4.2 Payment application must implement an automated audit trail to track and monitor access. PCI Data Security Standard Requirements 10.2 and 10.3

5. Develop secure payment applications

5.1 Develop all payment applications in accordance with PCI DSS (for example, secure authentication and logging) and based on industry best practices and incorporate information security throughout the software development life cycle. These processes must include the following: PCI Data Security Standard Requirement 6.3

5.1.1 Live PANS are not used for testing or development. PCI Data Security Standard Requirement 6.4.4.

- Validation of all input (to prevent cross-site scripting, injection flaws, malicious file execution, etc.)
- Validation of proper error handling
- Validation of secure cryptographic storage
- Validation of secure communications
- Validation of proper role-based access control (RBAC)

5.1.2 Separate development/test, and production environments

5.1.3 Removal of test data and accounts before production systems become active development. PCI Data Security Standard Requirement 6.4.4

5.1.4 Review of payment application code prior to release to customers after any significant change, to identify any potential coding vulnerability. Removal of custom payment application accounts, user IDs, and passwords before payment applications are released to customers

Note: This requirement for code reviews applies to all payment application components (both internal and public-facing web applications), as part of the system development life cycle required by PA-DSS Requirement 5.1 and PCI DSS Requirement 6.3. Code reviews can be conducted by knowledgeable internal personnel or third parties.

5.2 Develop all web payment applications (internal and external, and including web administrative access to product) based on secure coding guidelines such as the Open Web Application Security Project Guide. Cover prevention of common coding vulnerabilities in software development processes, to include:

- Injection flaws, with particular emphasis on SQL injection, Cross-site scripting (XSS) OS Command Injection, LDAP and Xpath injection flaws, as well as other injection flaws.
- Buffer Overflow.
- Insecure cryptographic storage.
- Insecure communications.
- Improper error handling.
- All HIGH vulnerabilities as identified in the vulnerability identification process at PA-DSS Requirement 7.1.
- Cross-site scripting (XSS)
- Improper access control such as insecure direct object references, failure to restrict URL access and directory traversal.
- Cross-site request forgery (CSRF)

Note: The vulnerabilities listed in PA-DSS Requirements 5.2.1 through 5.2.9 and in PCI DSS at 6.5.1 through 6.5.9 were current in the OWASP guide when PCI DSS v1.2 / PCI DSS v2.0 (01/01/10) were published. However, if and when the OWASP guide is updated, the current version must be used for these requirements.

5.3 Software vendor must follow change control procedures for all product software configuration changes. PCI Data Security Standard Requirement 6.4. 5.The procedures must include the following:

- Documentation of impact
- Management sign-off by appropriate parties
- Testing functionality to verify the new change(s) does not adversely impact the security of the system. Remove all testing configurations, samples, and data before finalizing the product for production.

- Back-out or product de-installation procedures

5.4 The payment application must not use or require use of unnecessary and insecure services and protocols (for example, NetBIOS, file-sharing, Telnet, unencrypted FTP must be secured via SSH, S-FTP, SSL, IPSec and other technology to implement end to end security). PCI Data Security Standard Requirement 2.2.2

6. Protect wireless transmissions

6.1 For payment applications using wireless technology, the wireless technology must be implemented securely. Payment applications using wireless technology must facilitate use of industry best practices (for example, IE↩ EE 802.11i) to implement strong encryption for authentication and transmission. Controls must be in place to protect the implemented wireless network from unknown wireless access points and clients. This includes testing the end users wireless deployment on a quarterly basis to detect unauthorized access points within the system. Change wireless vendor defaults, including but not limited to default wireless encryption keys, passwords, and S↩ NMP community strings. Maintain a detailed updated hardware list. The end to end wireless implementation must be end to end secure. The use of WEP as a security control was prohibited as of 30 June 2010. PCI Data Security Standard Requirements 1.2.3, 2.1.1, 4.1.1, 6.2, 11.1a-e and 11.4a-c.

7. Test payment applications to address vulnerabilities

7.1 Software vendors must establish a process to identify newly discovered security vulnerabilities (for example, subscribe to alert services freely available on the Internet) and to test their payment applications for vulnerabilities. Any underlying software or systems that are provided with or required by the payment application (for example, web servers, third-party libraries and programs) must be included in this process. Remove all test configurations, samples, and data after testing and before promoting the changes to production. PCI Data Security Standard Requirement 6.2

7.2 Software vendors must establish a process for timely development and deployment of security patches and upgrades, which includes delivery of updates and patches in a secure manner with a known chain-of-trust, and maintenance of the integrity of patch and update code during delivery and deployment.

8. Facilitate secure network implementation

8.1 The payment application must be able to be implemented into a secure network environment. Application must not interfere with use of devices, applications, or configurations required for PCI DSS compliance (for example, payment application cannot interfere with anti-virus protection, firewall configurations, or any other device, application, or configuration required for PCI DSS compliance). PCI Data Security Standard Requirements 1, 3, 4, 5, and 6.

9. Cardholder data must never be stored on a server connected to the Internet

9.1 The payment application must be developed such that the database server and web server are not required to be on the same server, nor is the database server required to be in the DMZ with the web server. PCI Data Security Standard Requirement 1.3.7

10. Facilitate secure remote software updates

10.1 If payment application updates are delivered securely via remote access into customers systems, software vendors must tell customers to turn on remote-access technologies only when needed for downloads from vendor

and to turn off immediately after download completes. Alternatively, if delivered via VPN or other high-speed connection, software vendors must advise customers to properly configure a firewall or a personal firewall product to secure authentication using a two factor authentication mechanism. PCI Data Security Standard Requirement 8.3

10.2 If payment application may be accessed remotely, remote access to the payment application must be authenticated using a two factor authentication mechanism. PCI Data Security Standard Requirement 8.3

10.3 Any remote access into the payment application must be done securely. If vendors, resellers/integrators, or customers can access customers payment applications remotely, the remote access must be implemented securely. PCI Data Security Standard Requirements 1, 8.3 and 12.3.9

11. Encrypt sensitive traffic over public networks

11.1 If the payment application sends, or facilitates sending, cardholder data over public networks, the payment application must support use of strong cryptography and security protocols such as SSL/TLS and Internet protocol security (IPSEC) to safeguard sensitive cardholder data during transmission over open, public networks. Examples of open, public networks that are in scope of the PCI DSS are: The Internet Wireless technologies Global System for Mobile Communications (GSM) General Packet Radio Service (GPRS) PCI Data Security Standard Requirement 4.1

11.2 The payment application must never send unencrypted PANs by end-user messaging technologies (for example, e-mail, instant messaging, and chat). PCI Data Security Standard Requirement 4.2

12. Encrypt all non-console administrative access

12.1 Instruct customers to encrypt all non-console administrative access using technologies such as SSH, VPN, or SSL/TLS for web-based management and other non-console administrative access. Telnet or remote login must never be used for administrative access. PCI Data Security Standard Requirement 2.3

13. Maintain instructional documentation and training programs for customers, resellers, and integrators

13.1 Develop, maintain, and disseminate a PA-DSS Implementation Guide(s) for customers, resellers, and integrators that accomplishes the following:

- Addresses all requirements in this document wherever the PA-DSS Implementation Guide is referenced.
- Includes a review at least annually and updates to keep the documentation current with all major and minor software changes as well as with changes to the requirements in this document.

13.2 Develop and implement training and communication programs to ensure payment application resellers and integrators know how to implement the payment application and related systems and networks according to the PA-DSS Implementation Guide and in a PCI DSS-compliant manner.

- Update the training materials on an annual basis and whenever new payment application versions are released.

2.5 More Information

IDTech Systems, Inc. highly recommends that merchants contact the card association(s) or their processing company and find out exactly what they mandate and/or recommend. Doing so may help merchants protect themselves from fines and fraud.

For more information related to security, visit:

- http://www.pcisecuritystandards.org
- http://www.visa.com/cisp
- http://www.sans.org/resources
- http://www.microsoft.com/security/default.asp
- https://sdp.mastercardintl.com/
- http://www.americanexpress.com/merchantspecs

CAPN questions: capninfocenter@aexp.com

Chapter 3

UniPay Main Transaction Commands

The methods below are provided as a reference to the main commands needed to execute an EMV transaction or perform a swipe.

3.1 EMV Methods

Start EMV Transaction

IDTechEMV::startEMVTransaction:timeout:transactionType:additionalTags:()

Begins an amount authorization request with the ICC. Returns authorization decision (approved, denied, or go online) in delegate method.

Complete Online EMV Transaction

IDTechEMV::completeOnlineEMVTransaction:resultCode:issuerAuthenticationData:issuerScripts:()

After receiving a host response, pass the result code as a string ("00"). The tags will be returned in the emv↩ TransactionData delegate protocol.

If there was a communication error with host, you must still finish the EMV transaction by passing "EMV_COMPL↩ ETION_RESULT_UNABLE_TO_GO_ONLINE".

```
typedef enum{
    EMV_COMPLETION_RESULT_ACCEPTED = 0X00,
    EMV_COMPLETION_RESULT_UNABLE_TO_GO_ONLINE = 0X01,
    EMV_COMPLETION_RESULT_TECHNICAL_ISSUE = 0X02,
    EMV_COMPLETION_RESULT_DECLINED = 0X03,
    EMV_COMPLETION_RESULT_ISSUER_REFERAL = 0X04
} EMV_COMPLETION_RESULT;
```

Terminal Configuration

IDTechEMV::retrieveTerminalData()
IDTechEMV::setTerminalData:()

Methods for terminal configuration. When setting the terminal data, you populate and pass and UniPay_Terminal↩ Data structure.

AID Management

IDTechEMV::retrieveApplicationData:()
IDTechEMV::removeApplicationData:()
IDTechEMV::setApplicationData:()
IDTechEMV::retrieveAIDList()

Methods for AID management. When setting the AID, you populate and pass UniPay_ApplicationID. When retrieving the AID list, the list of AID Names/length can be retrieved from the populated NSArray

Kernel Version

IDTechEMV::getEMVKernelVersion()

Method to retrieve kernel version.

APDU Communication

icc_exchangeAPDU:encrypted:response: (IDT_UniPay)

Allows the direct sending of APDU packets to ICC

3.2 MSR

Encryption Type

msr_setSwipeEncryption: (IDT_UniPay) msr_getSwipeEncryption: (IDT_UniPay)

Sets and gets the encrypted MSR Data Output Format.

Request Swipe

msr_startMSRSwipe: (IDT_UniPay)

Enables MSR to receive Swipe. Results are returned as IDTMSRData in swipeMSRData

```
- (void) swipeMSRData:(IDTMSRData*)cardData;
```

Cancel Swipe

msr_cancelMSRSwipe (IDT_UniPay)

Disables the MSR from receiving swipes.

Chapter 4

Core Implementation UniPay: iOS

IDTech Framework includes class libraries to interface with the UniPay. This guide assume a fair understanding of Xcode 5.0+ and general Apple iOS programming knowledge.

4.1 Integrating with IDTech framework

- Import the necessary framework/libraries
- Add Import statements to utilize frameworks
- Amend the view controller interface
- Implement optional delegate protocols
- Allocate/initialize UniPay objects
- Sample Project Tutorial

4.2 Import the necessary framework/libraries

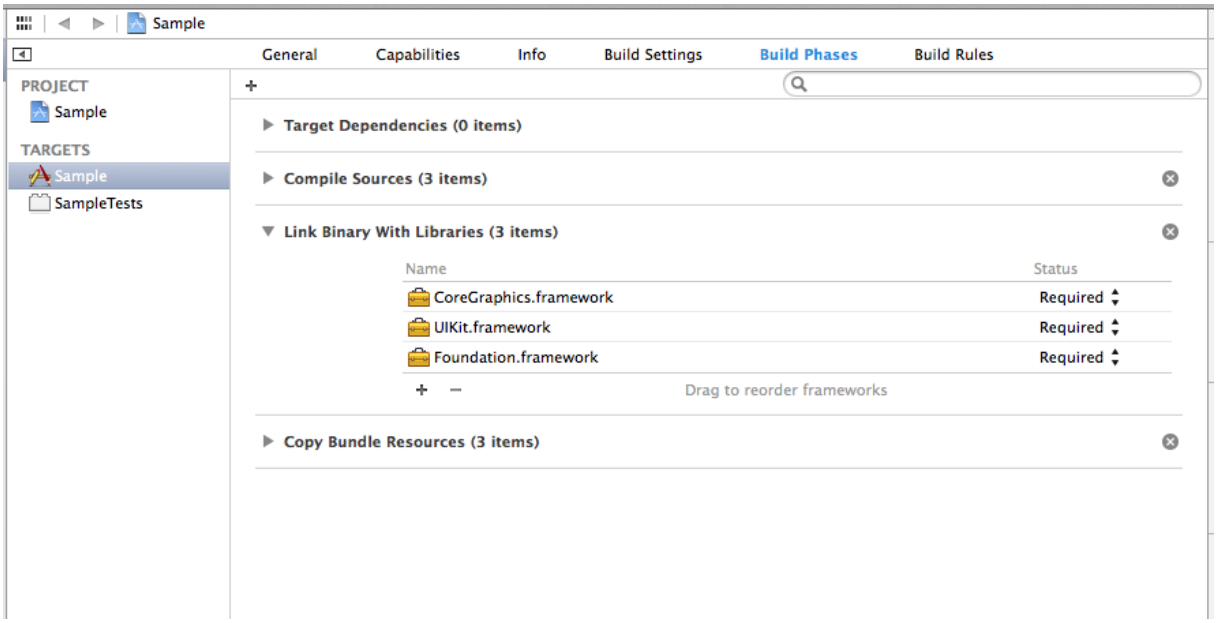
Communicating with UniPay requires the following framework/libraries to be imported into the project:

- IDTech.framework
- MediaPlayer.framework
- AVFoundation.framework
- AudioToolbox.framework
- CFNetwork.framework

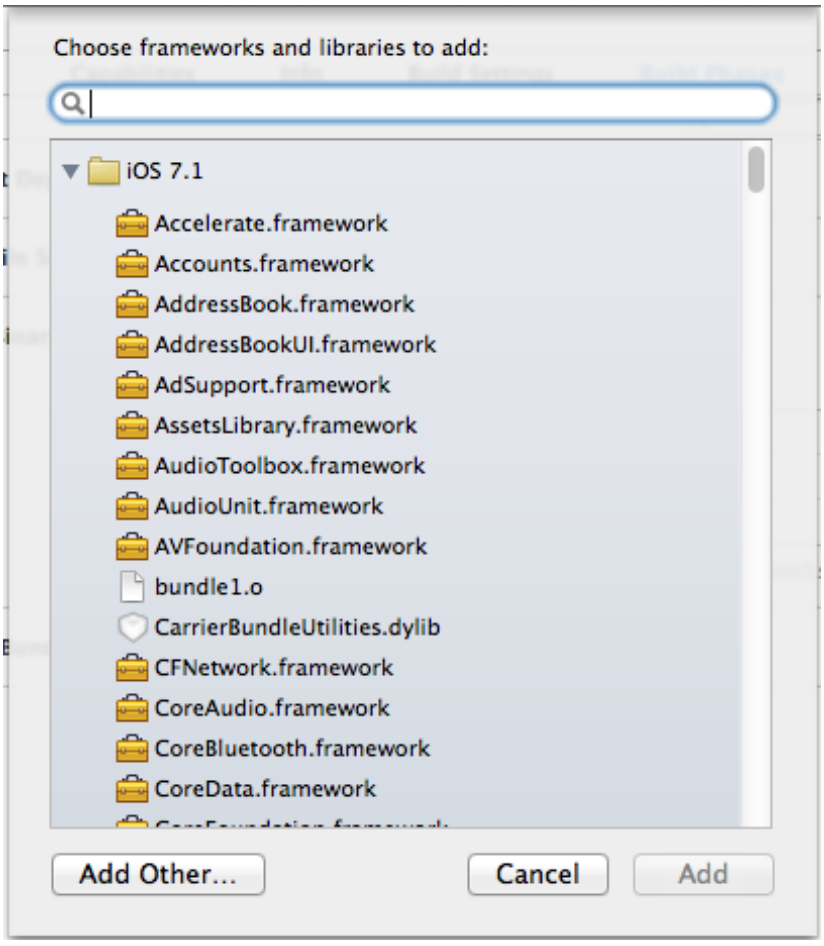
Also, import the following resource bundle:

- IDTech.bundle

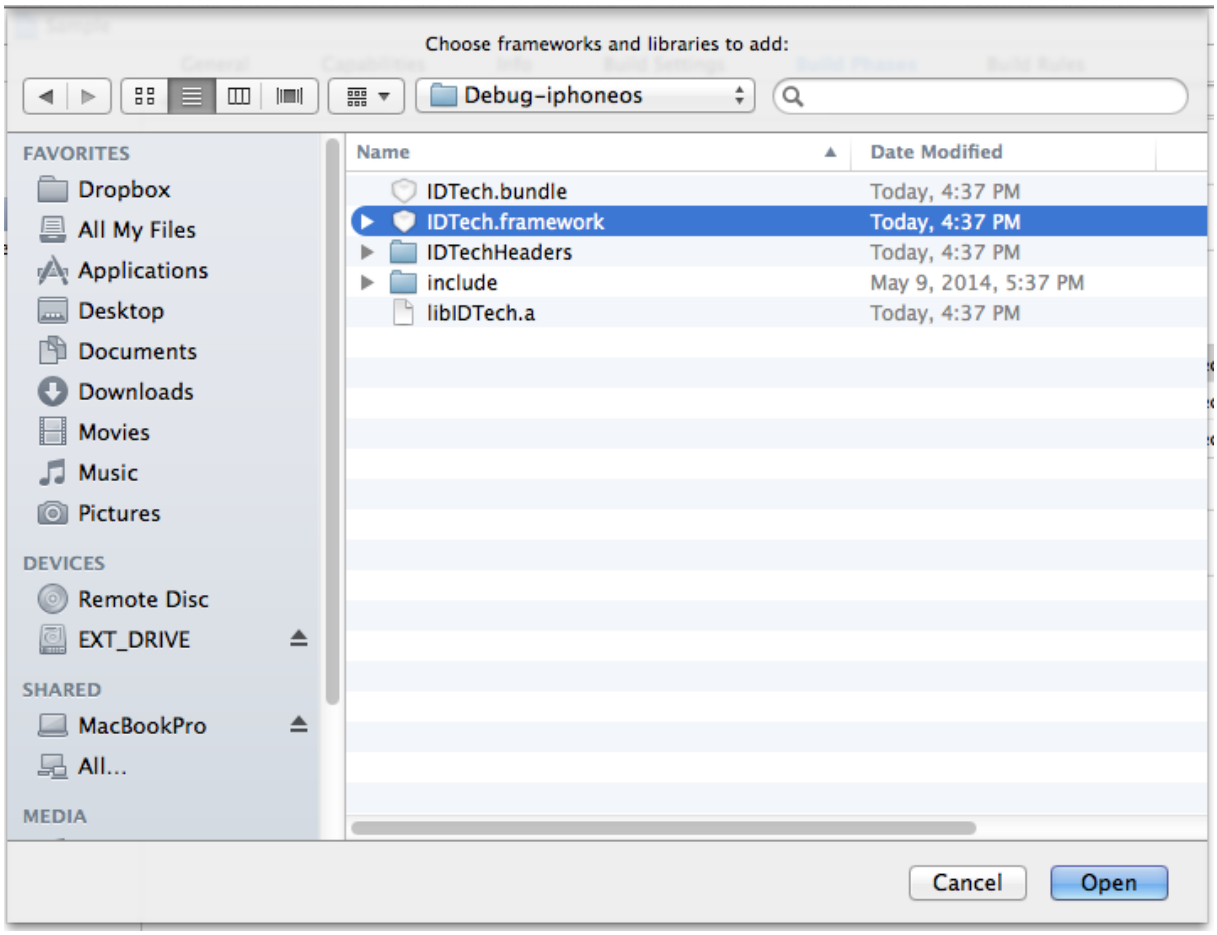
Under Build Phases, select Link Binary With Libraries and click the Add (+) button



On the Choose Frameworks screen, click "Add Other" in the lower left

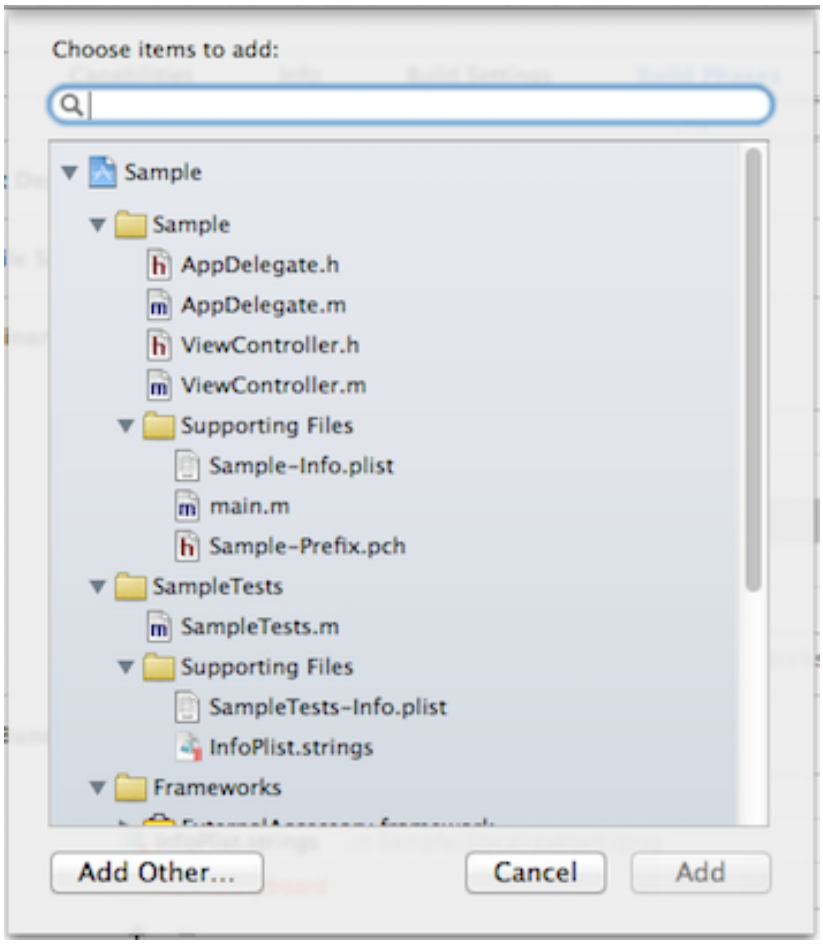


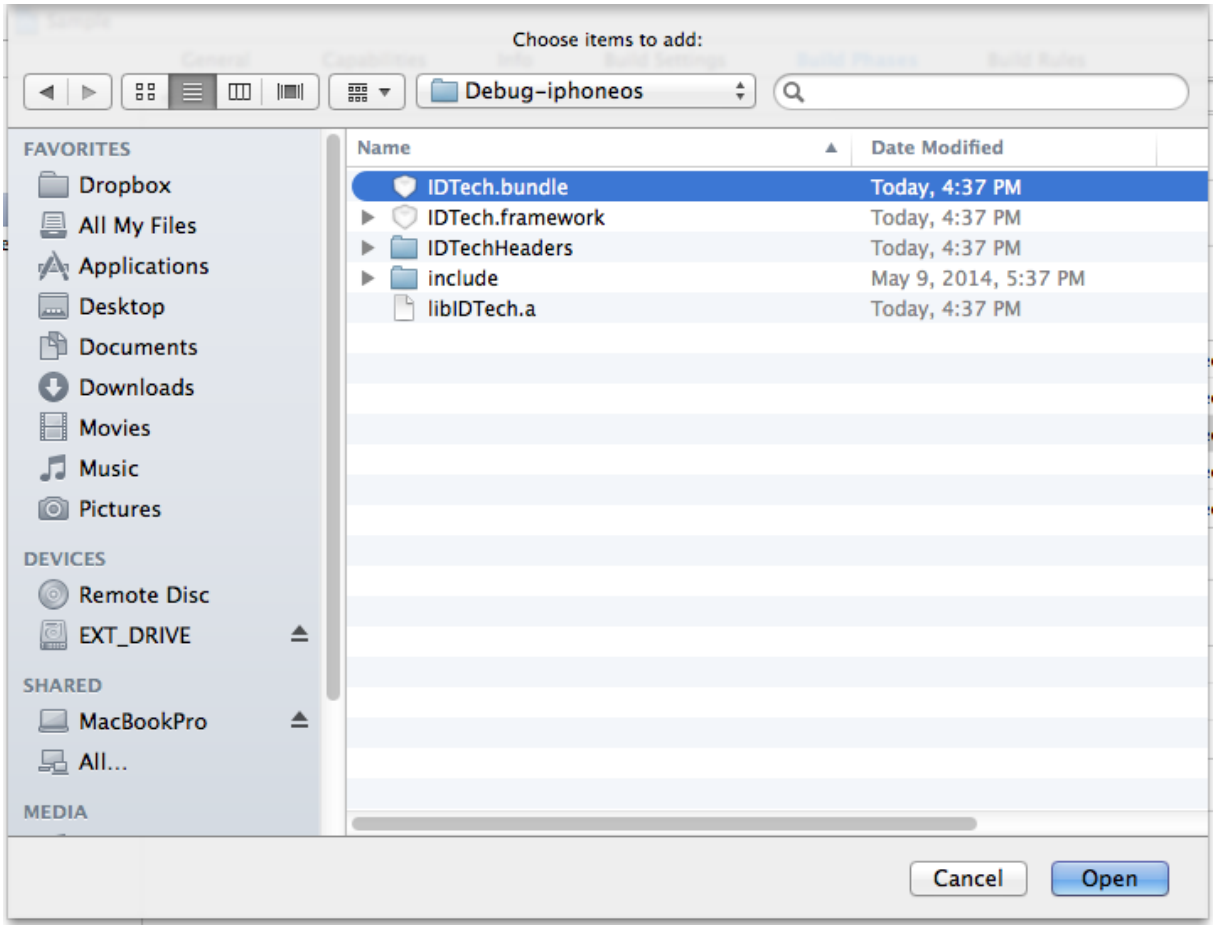
Navigate to the IDTech.framework folder, and click "Open"



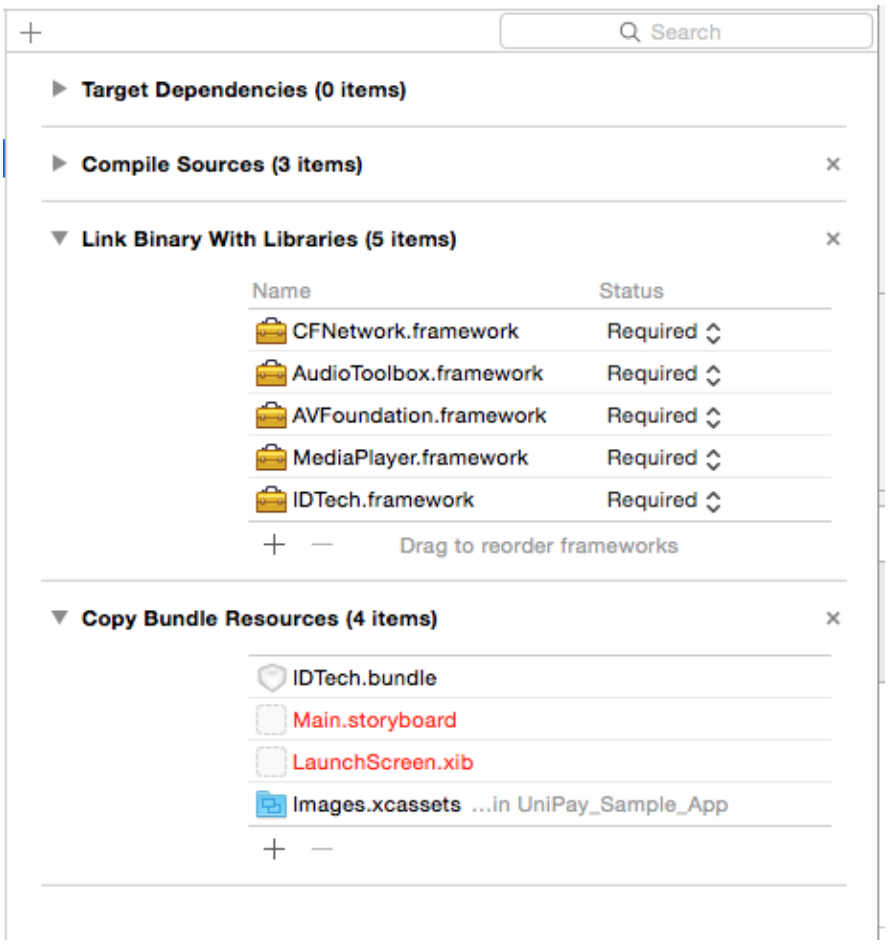
Repeat process for MediaPlayer.framework, AVFoundation.framework, AudioToolbox.framework, and CFNetwork.framework

Link another library. Under Copy Bundle, click the Add (+) button, click "Add Other", navigate to and select the IDTech.bundle file and click "Open"


Choose items to add:
Sample
Sample
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
Supporting Files
Sample-Info.plist
main.m
Sample-Prefix.pch
SampleTests
SampleTests.m
Supporting Files
SampleTests-Info.plist
InfoPlist.strings
Frameworks
Add Other...
Cancel
Add



The Build Phases should now include the required frameworks/libraries for the UniPay



4.3 Add Import statements to utilize frameworks

In the header files of the classes that will access IDTech Devices, use import statement utilize the frameworks:

```
#import <IDTech/IDTech.h>
```

4.4 Amend the view controller interface to include the framework delegate classes:

In the header files of the classes that will be a delegate of IDTech.framework, include the reference to the framework delegate class name:

```
@interface ViewController : UIViewController <IDT_UniPay_Delegate>
```

```
#import <UIKit/UIKit.h>
#import <IDTech/IDTech.h>


@interface ViewController : UIViewController <IDT_UniPay_Delegate>


@end
```

4.5 Implement any/all of the optional delegate protocols used to receive data from IDT_UniPay_Delegate:

```
-(void) deviceConnected;

-(void) deviceDisconnected;

- (void) plugStatusChange:(BOOL)deviceInserted;

- (void) dataInOutMonitor:(NSData*)data  incoming:(BOOL)isIncoming;

- (void) swipeMSRData:(IDTMSRData*)cardData;

- (void) deviceMessage:(NSString*)message;

-(void) eventFunctionICC: (Byte) nICC_Attached;
```

4.6 Call the Singleton instance of the IDT_UniPay framework object:

A Singleton instance has been established in the IDT_UniPay class. To utilize the delegate protocols, best practices would be initialize the connection by setting the delegate with the singleton instance.

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    //init object
    [IDT_UniPay sharedController].delegate = self;
}
```

4.7 Sample Project Tutorial

Using Xcode 5.0+, we will create a sample project that will interface with the UniPay and will perform the following activities:

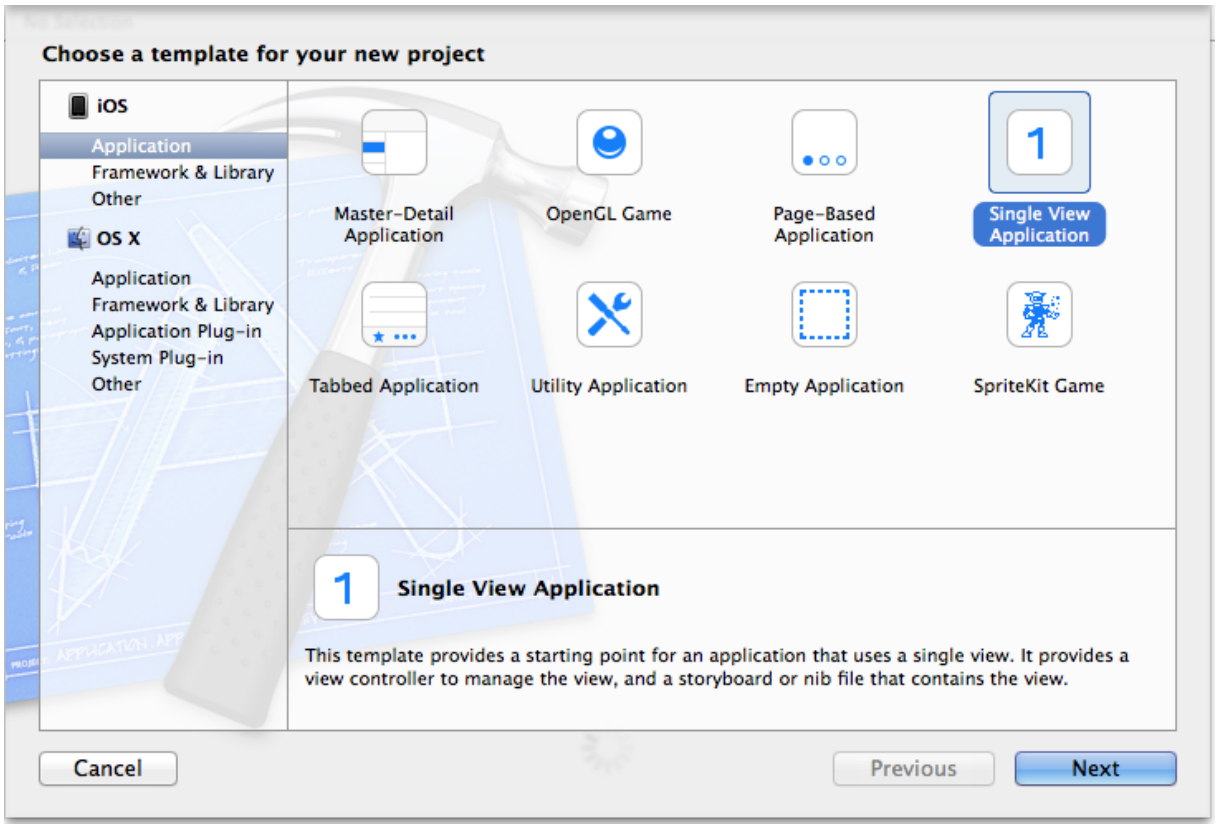
- Report Firmware
- Report connected/disconnected
- Turn on/off MSR reader and perform a card capture

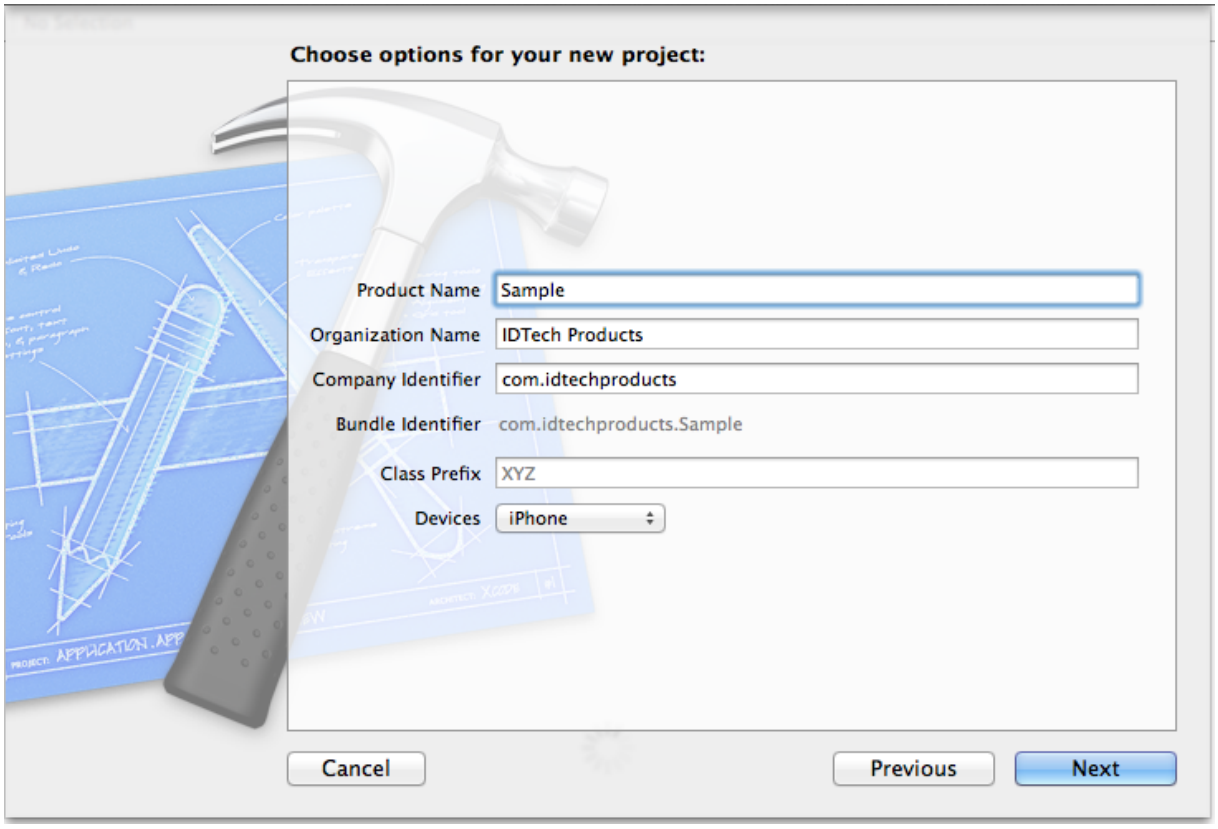
Protocol Delegates:

- Protocol to report unsolicited card swipes
- Protocol to report device connected
- Protocol to report device disconnected

4.7.1 Step 1: Create New Project

Create a new Single View Application in Xcode





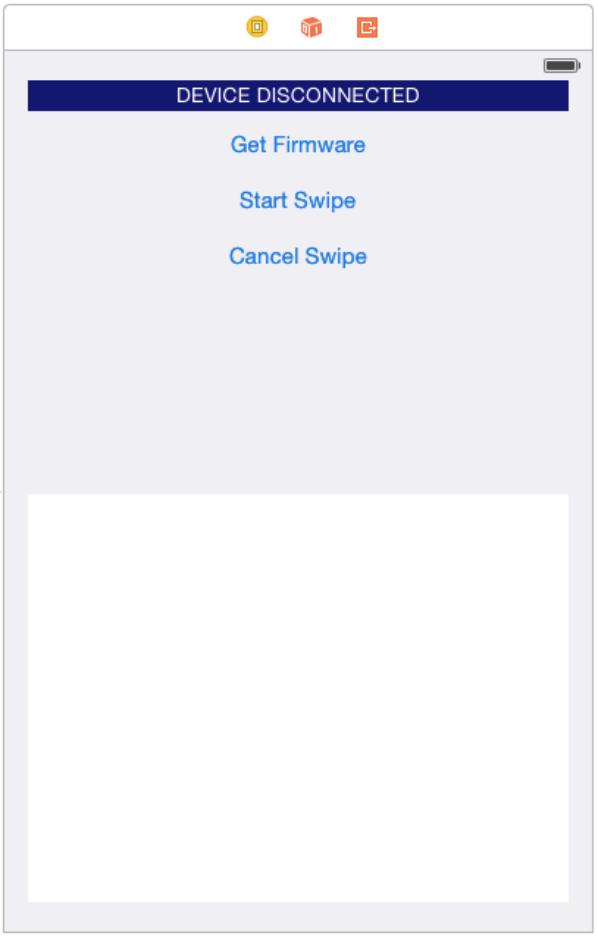
4.7.2 Step 2: Import Frameworks

Import the necessary framework/libraries

4.7.3 Step 3: Design Interface

Design the User Interface by editing the iPhone storyboard file
Open your storyboard and add items to so it contains the following buttons/fields:

- Add a label to the top that will signify connection/disconnection status.
- Add a text view to communicate data from the UniPay. Remove the Editable behavior if you don't want the keyboard to pop up if you accidentally select it.
- Add buttons to execute the following functions:
  - Get Firmware
  - Start Swipe
  - Cancel Swipe



4.7.4 Step 5: Configure Header File

In the header file, perform the following:

- Add Import statements to utilize frameworks

- Amend the view controller interface
- Create an IBOutlet for the UITextView and link it as a Referencing Outlet to the UITextView on the storyboard
- Create an IBOutlet for the UILabel and link it as a Referencing Outlet to the UILabel on the storyboard
- Create the 3 IBAction for the buttons, and link them to the "Touch Up Inside" event on the storyboard buttons

```
#import <UIKit/UIKit.h>
#import <IDTech/IDTech.h>


@interface ViewController : UIViewController <IDT_UniPay_Delegate>{
    IBOutlet UITextView *tv;
    IBOutlet UILabel *connectedLabel;
}
-(IBAction) msrON:(id)sender;
-(IBAction) msrOff:(id)sender;
-(IBAction) getFirmware:(id)sender;


@end
```

Storyboard Source Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB" version="3.0" toolsVersion="6250"
      systemVersion="14A389" targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES"
      useTraitCollections="YES" initialViewController="vXZ-lx-hvc">
    <dependencies>
        <deployment identifier="iOS"/>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="6244"/>
        <capability name="Alignment constraints with different attributes" minToolsVersion="5.1"/>
        <capability name="Constraints to layout margins" minToolsVersion="6.0"/>
    </dependencies>
    <scenes>
        <!--View Controller-->
        <scene sceneID="ufC-wZ-h7g">
            <objects>
                <viewController id="vXZ-lx-hvc" customClass="ViewController" sceneMemberID="viewController"
      >
                    <layoutGuides>
                        <viewControllerLayoutGuide type="top" id="jyV-Pf-zRb"/>
                        <viewControllerLayoutGuide type="bottom" id="2fi-mo-0CV"/>
                    </layoutGuides>
                    <view key="view" contentMode="scaleToFill" id="kh9-bI-dsS">
                        <rect key="frame" x="0.0" y="0.0" width="600" height="600"/>
                        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
                        <subviews>
                            <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
      horizontalHuggingPriority="251" verticalHuggingPriority="251" text="UNIPAY DISCONNECTED" textAlignment="center"
      lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
      translatesAutoresizingMaskIntoConstraints="NO" id="i1O-SW-UeA">
                                <rect key="frame" x="0.0" y="-21" width="42" height="21"/>
                                <color key="backgroundColor" white="0.0" alpha="1" colorSpace="
      calibratedWhite"/>
                                <fontDescription key="fontDescription" type="system" pointSize="13"/>
                                <color key="textColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
                                <nil key="highlightedColor"/>
                            </label>
                            <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
      center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
      translatesAutoresizingMaskIntoConstraints="NO" id="g4D-Lo-gdS">
                                <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                                <constraints>
                                    <constraint firstAttribute="height" constant="30" id="L4N-Q3-gDw"/>
                                </constraints>
                                <state key="normal" title="Get Firmware">
                                    <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
      calibratedWhite"/>
                                </state>
                                <variation key="default">
                                    <mask key="constraints">
                                        <exclude reference="L4N-Q3-gDw"/>
                                    </mask>
                                </variation>
                                <variation key="widthClass=compact">
                                    <mask key="constraints">
                                        <include reference="L4N-Q3-gDw"/>
                                    </mask>
                                </variation>
                                <connections>
                                    <action selector="getFirmware:" destination="vXZ-lx-hvc" eventType="
```

```
touchUpInside" id="LRv-H3-uBz"/>
                            </connections>
                        </button>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="qJM-Uo-OAl">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="mjm-6i-7SI"/>
                            </constraints>
                            <state key="normal" title="Start Swipe">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="mjm-6i-7SI"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="mjm-6i-7SI"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="msrON:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="6sg-MT-Toy"/>
                            </connections>
                        </button>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="e9C-sp-2x2">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="GEC-7y-53r"/>
                            </constraints>
                            <state key="normal" title="Cancel Swipe">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="GEC-7y-53r"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="GEC-7y-53r"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="msrOff:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="kCd-Zc-dgc"/>
                            </connections>
                        </button>
                        <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
scaleToFill" editable="NO" translatesAutoresizingMaskIntoConstraints="NO" id="cOU-wT-aWM">
                            <rect key="frame" x="0.0" y="0.0" width="240" height="128"/>
                            <color key="backgroundColor" white="1" alpha="1" colorSpace="
calibratedWhite"/>
                            <fontDescription key="fontDescription" type="system" pointSize="14"/>
                            <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
                        </textView>
                    </subviews>
                    <color key="backgroundColor" red="0.7662318441" green="0.94110946179999999" blue="
0.96664826770000001" alpha="1" colorSpace="calibratedRGB"/>
                    <constraints>
                        <constraint firstItem="g4D-Lo-gdS" firstAttribute="top" secondItem="i1O-SW-UeA"
 secondAttribute="bottom" constant="8" id="4JI-80-Dtl"/>
                        <constraint firstItem="e9C-sp-2x2" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="4ql-n9-Mob"/>
                        <constraint firstItem="cOU-wT-aWM" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="DkE-hg-tZJ"/>
                        <constraint firstItem="qJM-Uo-OAl" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="Gei-Yo-Nnl"/>
                        <constraint firstItem="qJM-Uo-OAl" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="MiH-76-1tD"/>
                        <constraint firstItem="g4D-Lo-gdS" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="atl-BT-IkW"/>
                        <constraint firstItem="cOU-wT-aWM" firstAttribute="top" secondItem="e9C-sp-2x2"
 secondAttribute="bottom" constant="122" id="bUd-oq-RSO"/>
                        <constraint firstItem="e9C-sp-2x2" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="dsT-QT-3pX"/>
                        <constraint firstItem="i1O-SW-UeA" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="fvJ-Jb-F4D"/>
                        <constraint firstItem="g4D-Lo-gdS" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="i8i-vS-GHG"/>
```

```
                        <constraint firstItem="cOU-wT-aWM" firstAttribute="trailing" secondItem="
      kh9-bI-dsS" secondAttribute="trailingMargin" id="kTP-Pr-rCR"/>
                        <constraint firstItem="e9C-sp-2x2" firstAttribute="top" secondItem="qJM-Uo-OAl"
       secondAttribute="bottom" constant="8" id="lOc-pC-2TS"/>
                        <constraint firstItem="qJM-Uo-OAl" firstAttribute="top" secondItem="g4D-Lo-gdS"
       secondAttribute="bottom" constant="8" id="o1P-I3-vHp"/>
                        <constraint firstItem="i1O-SW-UeA" firstAttribute="top" secondItem="jyV-Pf-zRb"
       secondAttribute="bottom" id="tmv-so-RdM"/>
                        <constraint firstItem="2fi-mo-0CV" firstAttribute="top" secondItem="cOU-wT-aWM"
       secondAttribute="bottom" constant="20" id="wPc-2Y-bol"/>
                        <constraint firstItem="i1O-SW-UeA" firstAttribute="trailing" secondItem="
      kh9-bI-dsS" secondAttribute="trailingMargin" id="zKQ-gV-BB6"/>
                    </constraints>
                    <variation key="default">
                        <mask key="subviews">
                            <exclude reference="i1O-SW-UeA"/>
                            <exclude reference="g4D-Lo-gdS"/>
                            <exclude reference="qJM-Uo-OAl"/>
                            <exclude reference="e9C-sp-2x2"/>
                            <exclude reference="cOU-wT-aWM"/>
                        </mask>
                        <mask key="constraints">
                            <exclude reference="fvJ-Jb-F4D"/>
                            <exclude reference="tmv-so-RdM"/>
                            <exclude reference="zKQ-gV-BB6"/>
                            <exclude reference="4JI-80-Dtl"/>
                            <exclude reference="atl-BT-IkW"/>
                            <exclude reference="i8i-vS-GHG"/>
                            <exclude reference="Gei-Yo-Nnl"/>
                            <exclude reference="MiH-76-1tD"/>
                            <exclude reference="o1P-I3-vHp"/>
                            <exclude reference="4ql-n9-Mob"/>
                            <exclude reference="dsT-QT-3pX"/>
                            <exclude reference="lOc-pC-2TS"/>
                            <exclude reference="DkE-hg-tZJ"/>
                            <exclude reference="bUd-oq-RSO"/>
                            <exclude reference="kTP-Pr-rCR"/>
                            <exclude reference="wPc-2Y-bol"/>
                        </mask>
                    </variation>
                    <variation key="widthClass=compact">
                        <mask key="subviews">
                            <include reference="i1O-SW-UeA"/>
                            <include reference="g4D-Lo-gdS"/>
                            <include reference="qJM-Uo-OAl"/>
                            <include reference="e9C-sp-2x2"/>
                            <include reference="cOU-wT-aWM"/>
                        </mask>
                        <mask key="constraints">
                            <include reference="fvJ-Jb-F4D"/>
                            <include reference="tmv-so-RdM"/>
                            <include reference="zKQ-gV-BB6"/>
                            <include reference="4JI-80-Dtl"/>
                            <include reference="atl-BT-IkW"/>
                            <include reference="i8i-vS-GHG"/>
                            <include reference="Gei-Yo-Nnl"/>
                            <include reference="MiH-76-1tD"/>
                            <include reference="o1P-I3-vHp"/>
                            <include reference="4ql-n9-Mob"/>
                            <include reference="dsT-QT-3pX"/>
                            <include reference="lOc-pC-2TS"/>
                            <include reference="DkE-hg-tZJ"/>
                            <include reference="bUd-oq-RSO"/>
                            <include reference="kTP-Pr-rCR"/>
                            <include reference="wPc-2Y-bol"/>
                        </mask>
                    </variation>
                </view>
                <connections>
                    <outlet property="connectedLabel" destination="i1O-SW-UeA" id="aq7-Us-Fcj"/>
                    <outlet property="tv" destination="cOU-wT-aWM" id="5kv-jl-7Yp"/>
                </connections>
            </viewController>
            <placeholder placeholderIdentifier="IBFirstResponder" id="x5A-6p-PRh" sceneMemberID="
      firstResponder"/>
        </objects>
        <point key="canvasLocation" x="205.5" y="385"/>
      </scene>
    </scenes>
</document>
```

4.7.5 Step 6: Configure Method File

In the header file, perform the following:

- set delegate and initialize IDT_UniPay singleton object in the `viewDidLoad` method. Reference: Call the Singleton instance of the IDT_UniPay framework object

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    [[IDT_UniPay sharedController] setDelegate:self];
}
```

- Implement protocol delegate IDT_UniPayDelegate::deviceDisconnected() and IDT_UniPayDelegate::deviceConnected() to monitor connect/disconnect events and modify our connection label upon change. Reference: Implement optional delegate protocols

```
-(void)deviceConnected{
    connectedLabel.text = @"UNIPAY CONNECTED";
}

-(void)deviceDisconnected{
    connectedLabel.text = @"UNIPAY DISCONNECTED";
}
```

- Implement protocol delegate plugStatusChanged:() to automatically connect to UniPay when a device is inserted

```
- (void) plugStatusChange:(BOOL)deviceInserted{
    if (deviceInserted) {
        [self appendMessageToResults: @"device Attached."];
        [self appendMessageToResults: @"Start Connect Task..."];
        [[IDT_UniPay sharedController] device_connectToAudioReader];
    }
    else{
        [self appendMessageToResults: @"device removed."];
    }
}
```

- Implement protocol delegate swipeMSRData:() to receive unsolicited card swipe data. Reference: Implement optional delegate protocols

```
-(void) appendMessageToResults:(NSString*) message{
    [tv setText:[NSString stringWithFormat:@"%@\n%@", tv.text, message]];
    [tv scrollRangeToVisible:NSMakeRange([tv.text length], 0)];
}

- (void) swipeMSRData:(IDTMSRData*)cardData{
    NSLog(@"--MSR event Received, Type: %d, data: %@", cardData.event, cardData.encTrack1);
    switch (cardData.event) {
        case EVENT_MSR_CARD_DATA:
        {
            switch (cardData.captureEncodeType) {
                case CAPTURE_ENCODE_TYPE_ISOABA:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"
      ISO/ABA"]];
                    break;
                case CAPTURE_ENCODE_TYPE_AAMVA:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"
      AA/MVA"]];
                    break;

                case CAPTURE_ENCODE_TYPE_Other:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"Other
      "]];
                    break;

                case CAPTURE_ENCODE_TYPE_Raw:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"Raw"]
      ];
                    break;

                default:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"
      UNKNOWN"]];

                    break;
            }

            [self appendMessageToResults:[NSString stringWithFormat:@"Full card data: %@", cardData.
      cardData]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Track 1: %@", cardData.track1]];
```

```
            [self appendMessageToResults:[NSString stringWithFormat:@"Track 2: %@", cardData.track2]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Track 3: %@", cardData.track3]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Length Track 1: %i", cardData.
        track1Length]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Length Track 2: %i", cardData.
        track2Length]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Length Track 3: %i", cardData.
        track3Length]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Encoded Track 1: %@", cardData.
        encTrack1.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Encoded Track 2: %@", cardData.
        encTrack2.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Encoded Track 3: %@", cardData.
        encTrack3.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Hash Track 1: %@", cardData.
        hashTrack1.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Hash Track 2: %@", cardData.
        hashTrack2.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"Hash Track 3: %@", cardData.
        hashTrack3.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"KSN: %@", cardData.KSN.description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"\nSessionID: %@",  cardData.sessionID
        .description]];
            [self appendMessageToResults:[NSString stringWithFormat:@"\nReader Serial Number: %@",
        cardData.RSN]];
            [self appendMessageToResults:[NSString stringWithFormat:@"\nRead Status: %2X",  cardData.
        readStatus]];

            NSLog(@"Track 1: %@", cardData.track1);
            NSLog(@"Track 2: %@", cardData.track2);
            NSLog(@"Track 3: %@", cardData.track3);
            NSLog(@"Encoded Track 1: %@", cardData.encTrack1.description);
            NSLog(@"Encoded Track 2: %@", cardData.encTrack2.description);
            NSLog(@"Encoded Track 3: %@", cardData.encTrack3.description);
            NSLog(@"Hash Track 1: %@", cardData.hashTrack1.description);
            NSLog(@"Hash Track 2: %@", cardData.hashTrack2.description);
            NSLog(@"Hash Track 3: %@", cardData.hashTrack3.description);
            NSLog(@"SessionID: %@", cardData.sessionID.description);
            NSLog(@"nReader Serial Number: %@", cardData.RSN);
            NSLog(@"Read Status: %2X", cardData.readStatus);
            NSLog(@"KSN: %@", cardData.KSN.description);

            return;
        }
            break;

        case EVENT_MSR_CANCEL_KEY:
        {
            [self appendMessageToResults:[NSString stringWithFormat:@"(Event) MSR Cancel Key received: %@",
         cardData.encTrack1]];
            return;
        }
            break;

        case EVENT_MSR_BACKSPACE_KEY:
        {
            [self appendMessageToResults:[NSString stringWithFormat:@"(Event) MSR Backspace Key received:
         %@", cardData.encTrack1]];
            return;
        }
            break;

        case EVENT_MSR_ENTER_KEY:
        {
            [self appendMessageToResults:[NSString stringWithFormat:@"(Event) MSR Enter Key received: %@",
        cardData.encTrack1]];
            return;
        }
            break;

        case EVENT_MSR_UNKNOWN:
        {
            [self appendMessageToResults:[NSString stringWithFormat:@"(Event) MSR unknown event, data: %@",
         cardData.encTrack1]];
            return;
        }
            break;

        default:
            break;
    }
}
```

- Implement the button press methods

```
-(IBAction) msrON:(id)sender{
```

```
    RETURN_CODE rt = [[IDT_UniPay sharedController] msr_startMSRSwipe:0];
    if(RETURN_CODE_DO_SUCCESS == rt){
        [self appendMessageToResults:@"EnableMSR: OK."];
    }else
    {
        [self appendMessageToResults: @"Enable MSR Failure" ];
    }

}
-(IBAction) msrOff:(id)sender{
    RETURN_CODE rt = [[IDT_UniPay sharedController] msr_cancelMSRSwipe];
    if(RETURN_CODE_DO_SUCCESS == rt){
        [self appendMessageToResults:@"DisableMSR: OK."];
    }else
    {
        [self appendMessageToResults: @"Disable MSR Error" ];
    }

}
-(IBAction) getFirmware:(id)sender{
    NSString *result;
    RETURN_CODE rt = [[IDT_UniPay sharedController]  device_getFirmwareVersion:&result];
    if (RETURN_CODE_DO_SUCCESS == rt)
    {
        [self appendMessageToResults: [NSString stringWithFormat:@"Get FM info:  %@", result]];
    }
    else{
        [self appendMessageToResults: @"Get Firmware Error" ];
    }

}
```

Chapter 5

Core Implementation IDTechEMV

IDTechEMV.framework is a Certified Level 2 EMV Kernel for UniPay. It is an add-on framework to the IDT_UniPay class. An existing UniPay project must be established before IDTechEMV can be incorporated with it.

5.1 Integrating with IDT_UniPay project

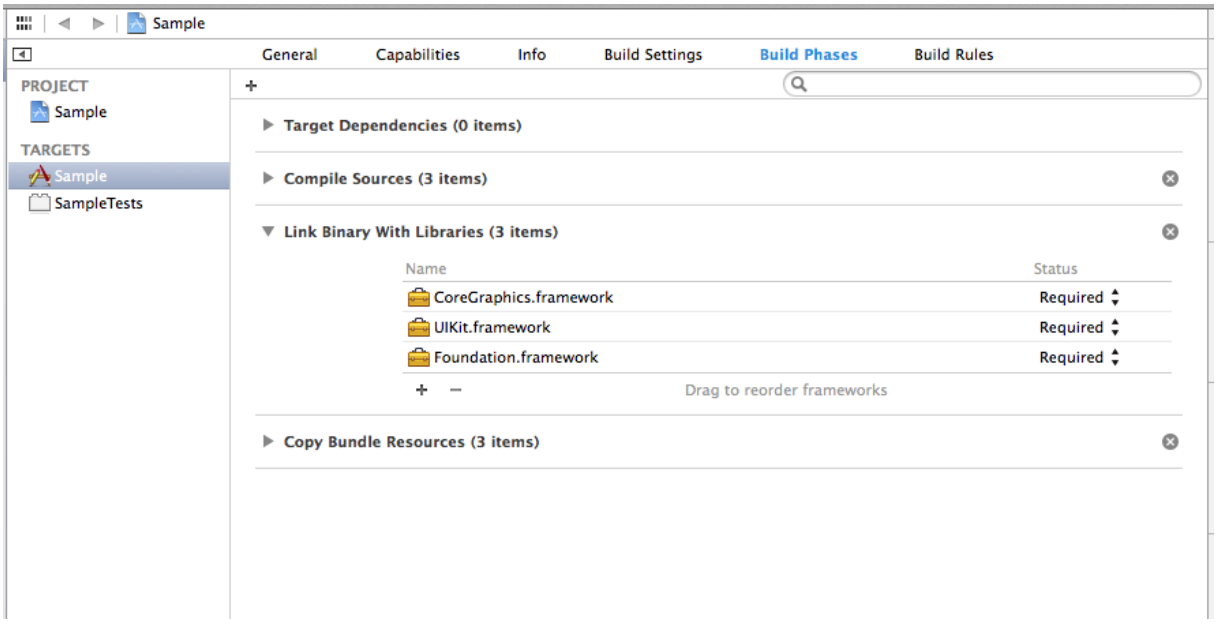
- Import the necessary framework/libraries
- Add Import statements to utilize frameworks
- Amend the view controller interface
- Implement optional delegate protocols
- Allocate/initialize IDT_BTPay objects
- Sample Project Tutorial

5.2 Import the necessary framework/libraries

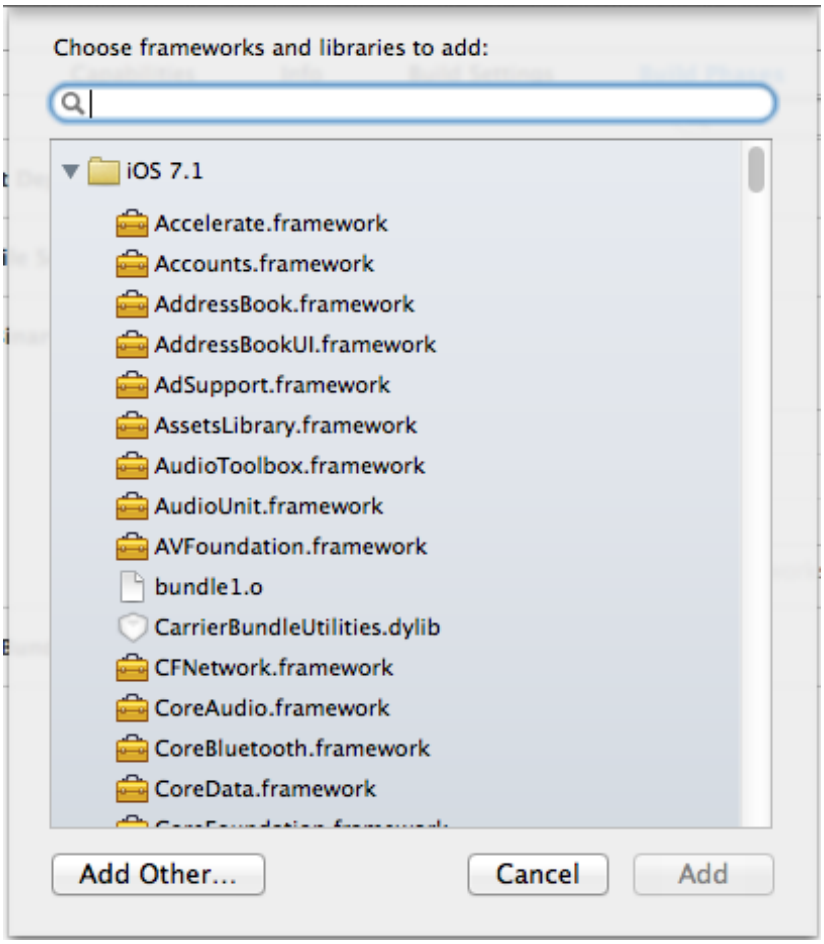
Adding EMV functionality to an existing UniPay project requires the following framework to be imported:

- IDTechEMV.framework

Under Build Phases, select Link Binary With Libraries and click the Add (+) button



On the Choose Frameworks screen, click "Add Other" in the lower left



Navigate to the IDTechEMV.framework folder, and click "Open"

5.3 Add Import statements to utilize frameworks

In the header files of the classes that will access the device, use import statement utilize the frameworks:

```
#import <IDTechEMV/IDTechEMV.h>
```

5.4 Amend the view controller interface to include the framework delegate classes:

In the header files of the classes that will be a delegate of UniPayEVM.framework, include the reference to the framework delegate class name:

```
@interface ViewController : UIViewController <IDT_UniPay_Delegate,IDTechEMV_Delegate>
```

```
#import <UIKit/UIKit.h>
#import <IDTech/IDTech.h>
#import <UniPayEMV/UniPayEMV.h>


@interface ViewController : UIViewController <IDT_UniPay_Delegate,UniPayEMV_Delegate>{
    IBOutlet UITextView *tv;
    IBOutlet UILabel *connectedLabel;
}
-(IBAction) msrON:(id)sender;
-(IBAction) msrOff:(id)sender;
-(IBAction) getFirmware:(id)sender;


@end
```

5.5 Implement any/all of the optional delegate protocols used to receive data from IDTechEMV.framework:

```
- (void) confirmApplicationSelection:(NSArray*)labelArray retry:(BOOL)tryAgain{
}

- (void) languagePreference:(NSData*)lang{
}

- (void) emvTransactionData:(IDTEMVData*)emvData errorCode:(int)error performReversal:(BOOL)reversal{
}

- (void) emvTransactionMessage:(MESSAGE_Types)message{
}

- (void) swipeMSRDataEMV:(IDTMSRData*)cardData  emv:(NSDictionary*)emvData{
}
```

5.6 Call the Singleton instance of the IDTechEMV framework object:

To use a device, a Singleton instance has been established in the IDTechEMV class. To utilize the delegate protocols, best practices would be initialize the connection by setting the delegate with the singleton instance.

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    [IDT_UniPay sharedController].delegate = self;
    [IDTechEMV sharedController].delegate = self;
}
```

5.7 Sample Project Tutorial

We will add EMV capabilities to the previous sample project tutorial for UniPay:

- start EMV transaction
- complete EMV transaction

Protocol Delegates:

- Protocol to report EMV card swipes
- Protocol to report EMV data

5.7.1 Step 1: Create iOS Sample Project

Create the sample project for UniPay

Sample Project Tutorial

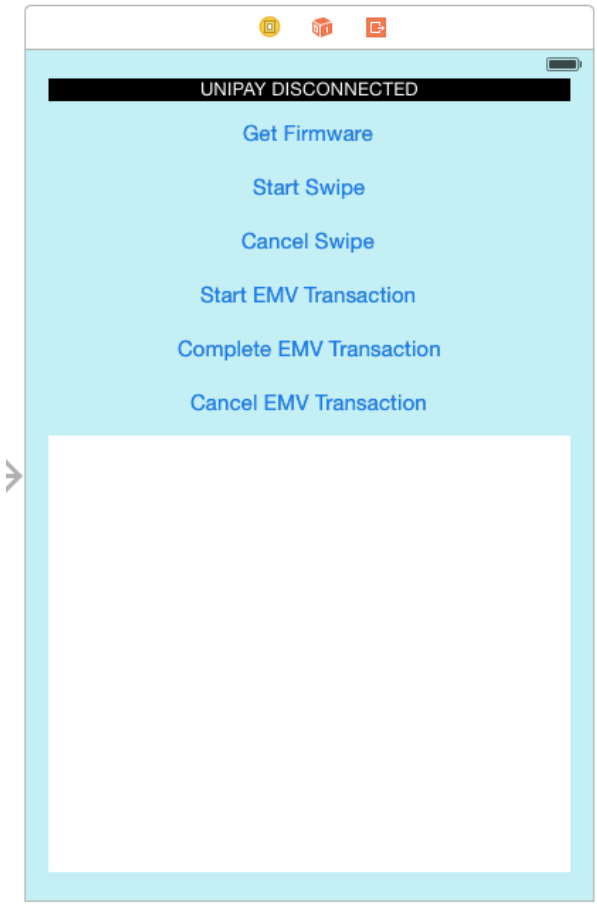
5.7.2 Step 2: Import Frameworks

Import the necessary framework/libraries

5.7.3 Step 3: Append Interface

Add to User Interface by editing the storyboard file
Open your main storyboard file, and add three more buttons:

- Add a button to start EMV transaction.
- Add a button to complete EMV transaction.
- Add a button to cancel emv transaction:



5.7.4 Step 4: Configure Header File

In the header file, perform the following:

- Add Import statements to utilize frameworks
- Amend the view controller interface
- Create the 3 IBAction for the buttons, and link them to the selector on the xib buttons

```
#import <UIKit/UIKit.h>
#import <IDTech/IDTech.h>
#import <IDTechEMV/IDTechEMV.h>


@interface ViewController : UIViewController <IDT_UniPay_Delegate,IDTechEMV_Delegate>{
    IBOutlet UITextView *tv;
    IBOutlet UILabel *connectedLabel;
}
-(IBAction) msrON:(id)sender;
-(IBAction) msrOff:(id)sender;
-(IBAction) getFirmware:(id)sender;
-(IBAction) startEMV:(id)sender;
-(IBAction) completeEMV:(id)sender;
-(IBAction) cancelEMV:(id)sender;


@end
```

XIB Source Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB" version="3.0" toolsVersion="6250"
      systemVersion="14A389" targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES"
```

```
  useTraitCollections="YES" initialViewController="vXZ-lx-hvc">
<dependencies>
    <deployment identifier="iOS"/>
    <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="6244"/>
    <capability name="Alignment constraints with different attributes" minToolsVersion="5.1"/>
    <capability name="Constraints to layout margins" minToolsVersion="6.0"/>
</dependencies>
<scenes>
    <!--View Controller-->
    <scene sceneID="ufC-wZ-h7g">
        <objects>
            <viewController id="vXZ-lx-hvc" customClass="ViewController" sceneMemberID="viewController"
  >
                <layoutGuides>
                    <viewControllerLayoutGuide type="top" id="jyV-Pf-zRb"/>
                    <viewControllerLayoutGuide type="bottom" id="2fi-mo-0CV"/>
                </layoutGuides>
                <view key="view" contentMode="scaleToFill" id="kh9-bI-dsS">
                    <rect key="frame" x="0.0" y="0.0" width="600" height="600"/>
                    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
                    <subviews>
                        <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
  horizontalHuggingPriority="251" verticalHuggingPriority="251" text="UNIPAY DISCONNECTED" textAlignment="center"
  lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
  translatesAutoresizingMaskIntoConstraints="NO" id="i1O-SW-UeA">
                            <rect key="frame" x="0.0" y="-21" width="42" height="21"/>
                            <color key="backgroundColor" white="0.0" alpha="1" colorSpace="
  calibratedWhite"/>
                            <fontDescription key="fontDescription" type="system" pointSize="13"/>
                            <color key="textColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
                            <nil key="highlightedColor"/>
                        </label>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
  center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
  translatesAutoresizingMaskIntoConstraints="NO" id="g4D-Lo-gdS">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="L4N-Q3-gDw"/>
                            </constraints>
                            <state key="normal" title="Get Firmware">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
  calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="L4N-Q3-gDw"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="L4N-Q3-gDw"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="getFirmware:" destination="vXZ-lx-hvc" eventType="
  touchUpInside" id="LRv-H3-uBz"/>
                            </connections>
                        </button>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
  center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
  translatesAutoresizingMaskIntoConstraints="NO" id="qJM-Uo-OAl">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="mjm-6i-7SI"/>
                            </constraints>
                            <state key="normal" title="Start Swipe">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
  calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="mjm-6i-7SI"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="mjm-6i-7SI"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="msrON:" destination="vXZ-lx-hvc" eventType="
  touchUpInside" id="6sg-MT-Toy"/>
                            </connections>
                        </button>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
  center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
  translatesAutoresizingMaskIntoConstraints="NO" id="e9C-sp-2x2">
```

```
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="GEC-7y-53r"/>
                            </constraints>
                            <state key="normal" title="Cancel Swipe">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="GEC-7y-53r"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="GEC-7y-53r"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="msrOff:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="kCd-Zc-dgc"/>
                            </connections>
                        </button>
                        <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
scaleToFill" editable="NO" translatesAutoresizingMaskIntoConstraints="NO" id="cOU-wT-aWM">
                            <rect key="frame" x="0.0" y="0.0" width="240" height="128"/>
                            <color key="backgroundColor" white="1" alpha="1" colorSpace="
calibratedWhite"/>
                            <fontDescription key="fontDescription" type="system" pointSize="14"/>
                            <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
                        </textView>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="Lno-54-uPL">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="jxp-gq-pgx"/>
                            </constraints>
                            <state key="normal" title="Start EMV Transaction">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="jxp-gq-pgx"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="jxp-gq-pgx"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="startEMV:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="13Z-YV-Jna"/>
                            </connections>
                        </button>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="8oa-Fa-Cr0">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
                            <constraints>
                                <constraint firstAttribute="height" constant="30" id="g23-Uy-8zX"/>
                            </constraints>
                            <state key="normal" title="Complete EMV Transaction">
                                <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
                            </state>
                            <variation key="default">
                                <mask key="constraints">
                                    <exclude reference="g23-Uy-8zX"/>
                                </mask>
                            </variation>
                            <variation key="widthClass=compact">
                                <mask key="constraints">
                                    <include reference="g23-Uy-8zX"/>
                                </mask>
                            </variation>
                            <connections>
                                <action selector="completeEMV:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="sFK-R4-fu6"/>
                            </connections>
                        </button>
                        <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="
center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="dZy-ey-pqm">
                            <rect key="frame" x="-23" y="-15" width="46" height="30"/>
```

```
                        <constraints>
                            <constraint firstAttribute="height" constant="30" id="umd-iR-zI9"/>
                        </constraints>
                        <state key="normal" title="Cancel EMV Transaction">
                            <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
                        </state>
                        <variation key="default">
                            <mask key="constraints">
                                <exclude reference="umd-iR-zI9"/>
                            </mask>
                        </variation>
                        <variation key="widthClass=compact">
                            <mask key="constraints">
                                <include reference="umd-iR-zI9"/>
                            </mask>
                        </variation>
                        <connections>
                            <action selector="cancelEMV:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="R4F-u7-jv3"/>
                        </connections>
                    </button>
                </subviews>
                <color key="backgroundColor" red="0.7662318441" green="0.94110946179999999" blue="
0.96664826770000001" alpha="1" colorSpace="calibratedRGB"/>
                <constraints>
                    <constraint firstItem="dZy-ey-pqm" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="2L0-nG-LIK"/>
                    <constraint firstItem="g4D-Lo-gdS" firstAttribute="top" secondItem="i1O-SW-UeA"
 secondAttribute="bottom" constant="8" id="4JI-80-Dtl"/>
                    <constraint firstItem="e9C-sp-2x2" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="4ql-n9-Mob"/>
                    <constraint firstItem="Lno-54-uPL" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="7Iq-0b-6tA"/>
                    <constraint firstItem="cOU-wT-aWM" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="DkE-hg-tZJ"/>
                    <constraint firstItem="8oa-Fa-Cr0" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="Fcg-Ny-8dU"/>
                    <constraint firstItem="qJM-Uo-OAl" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="Gei-Yo-Nnl"/>
                    <constraint firstItem="8oa-Fa-Cr0" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="JLi-BG-iUm"/>
                    <constraint firstItem="qJM-Uo-OAl" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="MiH-76-1tD"/>
                    <constraint firstItem="Lno-54-uPL" firstAttribute="top" secondItem="e9C-sp-2x2"
 secondAttribute="bottom" constant="8" id="Nca-eQ-6sS"/>
                    <constraint firstItem="dZy-ey-pqm" firstAttribute="top" secondItem="8oa-Fa-Cr0"
 secondAttribute="bottom" constant="8" id="V2B-L0-Dzc"/>
                    <constraint firstItem="g4D-Lo-gdS" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="atl-BT-IkW"/>
                    <constraint firstItem="cOU-wT-aWM" firstAttribute="top" secondItem="e9C-sp-2x2"
 secondAttribute="bottom" constant="122" id="bUd-oq-RSO"/>
                    <constraint firstItem="dZy-ey-pqm" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="ctt-g9-K00"/>
                    <constraint firstItem="e9C-sp-2x2" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="dsT-QT-3pX"/>
                    <constraint firstItem="i1O-SW-UeA" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="fvJ-Jb-F4D"/>
                    <constraint firstItem="g4D-Lo-gdS" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="i8i-vS-GHG"/>
                    <constraint firstItem="cOU-wT-aWM" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="kTP-Pr-rCR"/>
                    <constraint firstItem="e9C-sp-2x2" firstAttribute="top" secondItem="qJM-Uo-OAl"
 secondAttribute="bottom" constant="8" id="lOc-pC-2TS"/>
                    <constraint firstItem="Lno-54-uPL" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="nYS-LI-2HM"/>
                    <constraint firstItem="qJM-Uo-OAl" firstAttribute="top" secondItem="g4D-Lo-gdS"
 secondAttribute="bottom" constant="8" id="o1P-I3-vHp"/>
                    <constraint firstItem="8oa-Fa-Cr0" firstAttribute="top" secondItem="Lno-54-uPL"
 secondAttribute="bottom" constant="8" id="pCl-es-4vT"/>
                    <constraint firstItem="i1O-SW-UeA" firstAttribute="top" secondItem="jyV-Pf-zRb"
 secondAttribute="bottom" id="tmv-so-RdM"/>
                    <constraint firstItem="2fi-mo-0CV" firstAttribute="top" secondItem="cOU-wT-aWM"
 secondAttribute="bottom" constant="20" id="wPc-2Y-bol"/>
                    <constraint firstItem="i1O-SW-UeA" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="zKQ-gV-BB6"/>
                </constraints>
                <variation key="default">
                    <mask key="subviews">
                        <exclude reference="i1O-SW-UeA"/>
                        <exclude reference="g4D-Lo-gdS"/>
                        <exclude reference="qJM-Uo-OAl"/>
                        <exclude reference="e9C-sp-2x2"/>
                        <exclude reference="cOU-wT-aWM"/>
                        <exclude reference="Lno-54-uPL"/>
                        <exclude reference="8oa-Fa-Cr0"/>
                        <exclude reference="dZy-ey-pqm"/>
```

```
                        </mask>
                        <mask key="constraints">
                            <exclude reference="fvJ-Jb-F4D"/>
                            <exclude reference="tmv-so-RdM"/>
                            <exclude reference="zKQ-gV-BB6"/>
                            <exclude reference="4JI-80-Dtl"/>
                            <exclude reference="atl-BT-IkW"/>
                            <exclude reference="i8i-vS-GHG"/>
                            <exclude reference="Gei-Yo-Nnl"/>
                            <exclude reference="MiH-76-1tD"/>
                            <exclude reference="o1P-I3-vHp"/>
                            <exclude reference="4ql-n9-Mob"/>
                            <exclude reference="dsT-QT-3pX"/>
                            <exclude reference="lOc-pC-2TS"/>
                            <exclude reference="7Iq-0b-6tA"/>
                            <exclude reference="Nca-eQ-6sS"/>
                            <exclude reference="nYS-LI-2HM"/>
                            <exclude reference="Fcg-Ny-8dU"/>
                            <exclude reference="JLi-BG-iUm"/>
                            <exclude reference="pCl-es-4vT"/>
                            <exclude reference="2L0-nG-LIK"/>
                            <exclude reference="V2B-L0-Dzc"/>
                            <exclude reference="ctt-g9-K00"/>
                            <exclude reference="DkE-hg-tZJ"/>
                            <exclude reference="bUd-oq-RSO"/>
                            <exclude reference="kTP-Pr-rCR"/>
                            <exclude reference="wPc-2Y-bol"/>
                        </mask>
                    </variation>
                    <variation key="widthClass=compact">
                        <mask key="subviews">
                            <include reference="i1O-SW-UeA"/>
                            <include reference="g4D-Lo-gdS"/>
                            <include reference="qJM-Uo-OAl"/>
                            <include reference="e9C-sp-2x2"/>
                            <include reference="cOU-wT-aWM"/>
                            <include reference="Lno-54-uPL"/>
                            <include reference="8oa-Fa-Cr0"/>
                            <include reference="dZy-ey-pqm"/>
                        </mask>
                        <mask key="constraints">
                            <include reference="fvJ-Jb-F4D"/>
                            <include reference="tmv-so-RdM"/>
                            <include reference="zKQ-gV-BB6"/>
                            <include reference="4JI-80-Dtl"/>
                            <include reference="atl-BT-IkW"/>
                            <include reference="i8i-vS-GHG"/>
                            <include reference="Gei-Yo-Nnl"/>
                            <include reference="MiH-76-1tD"/>
                            <include reference="o1P-I3-vHp"/>
                            <include reference="4ql-n9-Mob"/>
                            <include reference="dsT-QT-3pX"/>
                            <include reference="lOc-pC-2TS"/>
                            <include reference="7Iq-0b-6tA"/>
                            <include reference="Nca-eQ-6sS"/>
                            <include reference="nYS-LI-2HM"/>
                            <include reference="Fcg-Ny-8dU"/>
                            <include reference="JLi-BG-iUm"/>
                            <include reference="pCl-es-4vT"/>
                            <include reference="2L0-nG-LIK"/>
                            <include reference="V2B-L0-Dzc"/>
                            <include reference="ctt-g9-K00"/>
                            <include reference="DkE-hg-tZJ"/>
                            <include reference="bUd-oq-RSO"/>
                            <include reference="kTP-Pr-rCR"/>
                            <include reference="wPc-2Y-bol"/>
                        </mask>
                    </variation>
                </view>
                <connections>
                    <outlet property="connectedLabel" destination="i1O-SW-UeA" id="aq7-Us-Fcj"/>
                    <outlet property="tv" destination="cOU-wT-aWM" id="5kv-jl-7Yp"/>
                </connections>
            </viewController>
            <placeholder placeholderIdentifier="IBFirstResponder" id="x5A-6p-PRh" sceneMemberID="
    firstResponder"/>
        </objects>
        <point key="canvasLocation" x="205.5" y="385"/>
    </scene>
</scenes>
</document>
```

- Implement the button press methods

```
-(IBAction)startEMV:(id)sender{
```

```
    [[IDTechEMV sharedController] startEMVTransaction:2.95 timeout:30 transactionType:0 additionalTags:nil]
      ;
}

-(IBAction)completeEMV:(id)sender{
    [[IDTechEMV sharedController]  completeOnlineEMVTransaction:EMV_COMPLETION_RESULT_ACCEPTED resultCode:
      @"00" issuerAuthenticationData:nil issuerScripts:nil];
}

-(IBAction) cancelEMV:(id)sender{
    [[IDTechEMV sharedController] cancelTransaction];
}
```

- Implement the EMV delegates

```
- (void) emvTransactionMessage:(MESSAGE_Types)message{
    switch (message) {
        case MESSAGE_INSERT_CARD:
            [self appendMessageToResults:@"PLEASE INSERT OR SWIPE CARD"];
            break;
        case MESSAGE_REMOVE_CARD:
            [self appendMessageToResults:@"PLEASE REMOVE CARD"];
            break;
        case MESSAGE_BAD_ICC:
            [self appendMessageToResults:@"CHIP READ ERROR-USE MSR"];
            break;
        case MESSAGE_TRANSACTION_CANCELLED:
            [self appendMessageToResults:@"TRANSACTION CANCELLED"];
            break;
        case MESSAGE_FALLBACK_FAILED:
            [self appendMessageToResults:@"FALLBACK FAILED"];
            break;
        case MESSAGE_USE_CHIP_READER:
            [self appendMessageToResults:@"MUST USE ICC READER"];
            break;
        case MESSAGE_PROCESSING:
            [self appendMessageToResults:@"PROCESSING"];
            break;
        case MESSAGE_READY:
            [self appendMessageToResults:@"UniPay Ready"];
            break;
        case MESSAGE_USE_MSR:
            [self appendMessageToResults:@"Please Swipe Card"];
            break;
        case MESSAGE_NOT_ACCEPTED:
            [self appendMessageToResults:@"Not Accepted"];
            break;


        default:
            break;
    }

}



- (void) swipeMSRDataEMV:(IDTMSRData*)cardData  emv:(NSDictionary*)emvData{
    switch (cardData.event) {
        case EVENT_MSR_CARD_DATA:
        {
            switch (cardData.captureEncodeType) {
                case CAPTURE_ENCODE_TYPE_ISOABA:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"
      ISO/ABA"]];
                    break;
                case CAPTURE_ENCODE_TYPE_AAMVA:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"
      AA/MVA"]];
                    break;

                case CAPTURE_ENCODE_TYPE_Other:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"Other
      "]];
                    break;

                case CAPTURE_ENCODE_TYPE_Raw:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"Raw"]
      ];
                    break;

                default:
                    [self appendMessageToResults:[NSString stringWithFormat:@"Encryption Type: %@", @"
      UNKNOWN"]];
```

```
                    break;
            }

            if (cardData.cardData.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      Full card data: %@", cardData.cardData]];
            if (cardData.track1 > 0) [self appendMessageToResults:[NSString stringWithFormat:@"Track 1: %@"
      , cardData.track1]];
            if (cardData.track2 > 0) [self appendMessageToResults:[NSString stringWithFormat:@"Track 2: %@"
      , cardData.track2]];
            if (cardData.track3 > 0) [self appendMessageToResults:[NSString stringWithFormat:@"Track 3: %@"
      , cardData.track3]];
            if (cardData.encTrack1.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      Encoded Track 1: %@", cardData.encTrack1.description]];
            if (cardData.encTrack2.length > 0)  [self appendMessageToResults:[NSString stringWithFormat:@"
      Encoded Track 2: %@", cardData.encTrack2.description]];
            if (cardData.encTrack3.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      Encoded Track 3: %@", cardData.encTrack3.description]];
            if (cardData.hashTrack1.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      Hash Track 1: %@", cardData.hashTrack1.description]];
            if (cardData.hashTrack2.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      Hash Track 2: %@", cardData.hashTrack2.description]];
            if (cardData.hashTrack3.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      Hash Track 3: %@", cardData.hashTrack3.description]];
            if (cardData.KSN.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"KSN: %@"
      , cardData.KSN.description]];
            if (cardData.sessionID.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"
      \nSessionID: %@",  cardData.sessionID.description]];
            if (cardData.RSN.length > 0) [self appendMessageToResults:[NSString stringWithFormat:@"\nReader
       Serial Number: %@",  cardData.RSN]];
            [self appendMessageToResults:[NSString stringWithFormat:@"\nRead Status: %2X",  cardData.
      readStatus]];

            if (emvData != nil) {
                [self appendMessageToResults:emvData.description];
            }
            return;
        }
            break;


        default:
            break;
    }
    [self appendMessageToResults:[NSString stringWithFormat:@"Tags: %@",emvData.description]];
}



- (void) emvTransactionData:(IDTEMVData*)emvData errorCode:(int)error performReversal:(BOOL)reversal{
        [self appendMessageToResults:[NSString stringWithFormat:@"Result code = %2X, Error Code %4X",
      emvData.resultCode,error]];

    Byte *b = malloc(1);
    NSData*CID = [emvData.unencryptedTags objectForKey:@"9F27"];
    [CID getBytes:b];
    if (b[0] == 0x01) {

        [self appendMessageToResults:@"ERROR: SERVICE NOT ALLOWED"];
        free(b);
        return;
    }

    if (reversal) {
        [self appendMessageToResults:@"WARNING: TRANSACTIONS WAS APPROVED ONLINE, BUT TERMINAL DECLINED.
       PLEASE REVERSE/VOID ONLINE APPROVAL FOR THIS TRANSACTION"];
    }

    bool isError = (error != 0x100);
    if (!isError || emvData.resultCode == 2) {

        [self appendMessageToResults:@"GENERATE AC RESULTS:"];


    }
    else {
        [self appendMessageToResults:@"ERROR: TRANSACTION TERMINATED"];

    }


    if (emvData.unencryptedTags != nil) {
        [self appendMessageToResults:[NSString stringWithFormat:@"Tags:\n%@",emvData.unencryptedTags.
      description]];
    }

    if ((emvData.resultCode == 2) || (b[0] == 0x88) || (b[0] == 0x8B) || (b[0] == 0x8A)) {
        [self appendMessageToResults:@"GO ONLINE"];
```

```
        return;
    }
    else{
        NSString* cvmString = [(NSData*)[emvData.unencryptedTags objectForKey:@"9F34"] description];

        NSString* results = @"SUCCESS - TC";
        if ([CID.description hasPrefix:@"<0"]) {
            results = @"DECLINE - AAC";
        }


        [self appendMessageToResults:[NSString stringWithFormat:@"RESULT: %@\nTVR: %@\nTSI: %@\nCVM: %@",
      results,[(NSData*)[emvData.unencryptedTags objectForKey:@"95"] description],[(NSData*)[emvData.unencryptedTags
       objectForKey:@"9B"] description],cvmString]];

    }
    free(b);
}
```

Chapter 6

Enumeration Reference

IDTechEMV

```
typedef enum{
    EMV_RC_APPROVED = 0X00,
    EMV_RC_DECLINED = 0X01,
    EMV_RC_GO_ONLINE = 0X02,
    EMV_RC_FAILED = 0X03,
    EMV_RC_SYSTEM_ERROR = 0X05,
    EMV_RC_NOT_ACCEPT = 0X07,
    EMV_RC_FALLBACK = 0X0A,
    EMV_RC_CANCEL = 0X0C,
    EMV_RC_OTHER_ERROR = 0X0F,
    EMV_RC_TIME_OUT = 0X0D,
    EMV_RC_OFFLINE_APPROVED = 0X10,
    EMV_RC_OFFLINE_DECLINED = 0X11,
    EMV_RC_REFERRAL_PROCESSING = 0X12,
    EMV_RC_ERROR_APP_PROCESSING = 0X13,
    EMV_RC_ERROR_APP_READING = 0X14,
    EMV_RC_ERROR_DATA_AUTH = 0X15,
    EMV_RC_ERROR_PROCESSING_RESTRICTIONS = 0X16,
    EMV_RC_ERROR_CVM_PROCESSING = 0X17,
    EMV_RC_ERROR_RISK_MGMT = 0X18,
    EMV_RC_ERROR_TERM_ACTION_ANALYSIS = 0X19,
    EMV_RC_ERROR_CARD_ACTION_ANALYSIS = 0X1A,
    EMV_RC_ERROR_APP_SELECTION_TIMEOUT = 0X1B,
    EMV_RC_ERROR_NO_CARD_INSERTED = 0X1C,
    EMV_RC_ERROR_APP_SELECTING = 0X1D,
    EMV_RC_ERROR_READING_CARD_APP = 0X1E,
    EMV_RC_ERROR_POWER_CARD_ERROR = 0X1F,
    EMV_RC_ERROR_NO_RESULT_CODE_PROVIDED_FOR_COMPLETION = 0X20,
    EMV_RC_APPROVED_WITH_ADVISE_NO_REASON = 0X21,
    EMV_RC_APPROVED_WITH_ADVISE_IA_FAILED = 0X22,
    EMV_RC_ERROR_AMOUNT_NOT_SPECIFIED = 0X23
} EMV_RC_Types;
```

```
typedef enum{
    MESSAGE_INSERT_CARD = 0,
    MESSAGE_REMOVE_CARD,
    MESSAGE_BAD_ICC,
    MESSAGE_TRANSACTION_CANCELLED,
    MESSAGE_FALLBACK_FAILED,
    MESSAGE_USE_CHIP_READER,
    MESSAGE_PROCESSING,
    MESSAGE_READY,
    MESSAGE_USE_MSR,
    MESSAGE_NOT_ACCEPTED
}MESSAGE_Types;
```

```
typedef enum{
    EMV_COMPLETION_RESULT_ACCEPTED = 0X00,
    EMV_COMPLETION_RESULT_UNABLE_TO_GO_ONLINE = 0X01,
```

```
    EMV_COMPLETION_RESULT_TECHNICAL_ISSUE = 0X02,
    EMV_COMPLETION_RESULT_DECLINED = 0X03,
    EMV_COMPLETION_RESULT_ISSUER_REFERAL = 0X04
} EMV_COMPLETION_RESULT;
```

IDTMSRData

```
typedef enum _CAPTURE_ENCODE_TYPE{
    CAPTURE_ENCODE_TYPE_ISOABA=0,
    CAPTURE_ENCODE_TYPE_AAMVA=1,
    CAPTURE_ENCODE_TYPE_Other=3,
    CAPTURE_ENCODE_TYPE_Raw=4,
    CAPTURE_ENCODE_TYPE_JIS_II=5,
    CAPTURE_ENCODE_TYPE_JIS_I=6,
    CAPTURE_ENCODE_TYPE_MANUAL_ENTRY=7
} CAPTURE_ENCODE_TYPE;
```

```
typedef enum{
    CAPTURE_ENCRYPT_TYPE_TDES=0,
    CAPTURE_ENCRYPT_TYPE_AES=1
} CAPTURE_ENCRYPT_TYPE;
```

IDTCommon

```
typedef enum{
    POWER_ON_OPTION_IFS_FLAG=1,
    POWER_ON_OPTION_EXPLICIT_PPS_FLAG=2,
    POWER_ON_OPTION_AUTO_PPS_FLAG=64,
    POWER_ON_OPTION_IFS_RESPONSE_CHECK_FLAG=128
}POWER_ON_OPTION;
```

```
typedef enum{
    LANGUAGE_TYPE_ENGLISH=1,
    LANGUAGE_TYPE_PORTUGUESE,
    LANGUAGE_TYPE_SPANISH,
    LANGUAGE_TYPE_FRENCH
}LANGUAGE_TYPE;
```

```
typedef enum{
    PIN_KEY_TDES_MKSK_extp=0x00,
    PIN_KEY_TDES_DUKPT_extp=0x01,
    PIN_KEY_TDES_MKSK_intl=0x10,
    PIN_KEY_TDES_DUKPT_intl=0x11,
}PIN_KEY_Types;
```

```
typedef enum{
    EVENT_PINPAD_UNKNOWN = 11,
    EVENT_PINPAD_ENCRYPTED_PIN,
    EVENT_PINPAD_NUMERIC,
    EVENT_PINPAD_AMOUNT,
    EVENT_PINPAD_ACCOUNT,
    EVENT_PINPAD_ENCRYPTED_DATA,
    EVENT_PINPAD_CANCEL,
    EVENT_PINPAD_TIMEOUT,
    EVENT_PINPAD_FUNCTION_KEY,
    EVENT_PINPAD_DATA_ERROR
}EVENT_PINPAD_Types;
```

```
typedef enum{
    EVENT_MSR_UNKNOWN = 31,
    EVENT_MSR_CARD_DATA,
    EVENT_MSR_CANCEL_KEY,
    EVENT_MSR_BACKSPACE_KEY,
    EVENT_MSR_ENTER_KEY,
    EVENT_MSR_DATA_ERROR,
    EVENT_MSR_ICC_START,
    EVENT_BTPAY_CARD_DATA,
    EVENT_UNIPAYII_EMV_NO_ICC_MSR_DATA,
    EVENT_UNIPAYII_EMV_FALLBACK_DATA
}EVENT_MSR_Types;
```

```
typedef enum{
    EVENT_ACTIVE_TRANSACTION = 51
}EVENT_CTLS_Types;
```

```
typedef enum {
    RETURN_CODE_DO_SUCCESS = 0,
    RETURN_CODE_ERR_DISCONNECT,
    RETURN_CODE_ERR_CMD_RESPONSE,
    RETURN_CODE_ERR_TIMEDOUT,
    RETURN_CODE_ERR_INVALID_PARAMETER,
    RETURN_CODE_SDK_BUSY_MSR,
    RETURN_CODE_SDK_BUSY_PINPAD,
    RETURN_CODE_SDK_BUSY_CTLS,
    RETURN_CODE_ERR_OTHER,
    RETURN_CODE_FAILED,
    RETURN_CODE_NOT_ATTACHED,
    RETURN_CODE_MONO_AUDIO,
    RETURN_CODE_CONNECTED,
    RETURN_CODE_LOW_VOLUME,
    RETURN_CODE_CANCELED,


    RETURN_CODE_EMV_AUTHORIZATION_ACCEPTED = 0x0E00,
    RETURN_CODE_EMV_AUTHORIZATION_UNABLE_TO_GO_ONLINE = 0x0E01,
    RETURN_CODE_EMV_AUTHORIZATION_TECHNICAL_ISSUE = 0x0E02,
    RETURN_CODE_EMV_AUTHORIZATION_DECLINED = 0x0E03,
    RETURN_CODE_EMV_AUTHORIZATION_ISSUER_REFERRAL = 0x0E04,

    RETURN_CODE_EMV_APPROVED = 0x0F00,ction
    RETURN_CODE_EMV_DECLINED = 0x0F01,
    RETURN_CODE_EMV_GO_ONLINE = 0x0F02,
    RETURN_CODE_EMV_FAILED = 0x0F03,
    RETURN_CODE_EMV_SYSTEM_ERROR = 0x0F05,
    RETURN_CODE_EMV_NOT_ACCEPTED = 0x0F07,
    RETURN_CODE_EMV_FALLBACK = 0x0F0A,
    RETURN_CODE_EMV_CANCEL = 0x0F0C,
    RETURN_CODE_EMV_TIMEOUT = 0x0F0D,
    RETURN_CODE_EMV_OTHER_ERROR = 0x0F0F,
    RETURN_CODE_EMV_OFFLINE_APPROVED = 0x0F10,
    RETURN_CODE_EMV_OFFLINE_DECLINED = 0x0F11,



    RETURN_CODE_EMV_NEW_SELECTION = 0x0F21,
    RETURN_CODE_EMV_NO_AVAILABLE_APPS = 0x0F22,
    RETURN_CODE_EMV_NO_TERMINAL_FILE = 0x0F23,
    RETURN_CODE_EMV_NO_CAPK_FILE = 0x0F24,
    RETURN_CODE_EMV_NO_CRL_ENTRY = 0x0F25,
    RETURN_CODE_BLOCKING_DISABLED = 0x0FFE,
    RETURN_CODE_COMMAND_UNAVAILABLE = 0x0FFF


} RETURN_CODE;
```

```
typedef enum{
    EMV_RESULT_CODE_APPROVED = 0X00,
    EMV_RESULT_CODE_DECLINED = 0X01,
    EMV_RESULT_CODE_GO_ONLINE = 0X02,
```

```
    EMV_RESULT_CODE_FAILED = 0X03,
    EMV_RESULT_CODE_SYSTEM_ERROR = 0X05,
    EMV_RESULT_CODE_NOT_ACCEPT = 0X07,
    EMV_RESULT_CODE_FALLBACK = 0X0A,
    EMV_RESULT_CODE_CANCEL = 0X0C,
    EMV_RESULT_CODE_OTHER_ERROR = 0X0F,
    EMV_RESULT_CODE_TIME_OUT = 0X0D,
    EMV_RESULT_CODE_OFFLINE_APPROVED = 0X10,
    EMV_RESULT_CODE_OFFLINE_DECLINED = 0X11,
    EMV_RESULT_CODE_REFERRAL_PROCESSING = 0X12,
    EMV_RESULT_CODE_ERROR_APP_PROCESSING = 0X13,
    EMV_RESULT_CODE_ERROR_APP_READING = 0X14,
    EMV_RESULT_CODE_ERROR_DATA_AUTH = 0X15,
    EMV_RESULT_CODE_ERROR_PROCESSING_RESTRICTIONS = 0X16,
    EMV_RESULT_CODE_ERROR_CVM_PROCESSING = 0X17,
    EMV_RESULT_CODE_ERROR_RISK_MGMT = 0X18,
    EMV_RESULT_CODE_ERROR_TERM_ACTION_ANALYSIS = 0X19,
    EMV_RESULT_CODE_ERROR_CARD_ACTION_ANALYSIS = 0X1A,
    EMV_RESULT_CODE_ERROR_APP_SELECTION_TIMEOUT = 0X1B,
    EMV_RESULT_CODE_ERROR_DATA_LEN_INCORRECT = 0X1C,
    EMV_RESULT_CODE_CALL_YOUR_BANK = 0X1D,
    EMV_RESULT_CODE_NO_ICC_ON_CARD = 0X1E,
    EMV_RESULT_CODE_NEW_SELECTION = 0X1F,
    EMV_RESULT_CODE_START_TRANSACTION_SUCCESS = 0X20
} EMV_RESULT_CODE_Types;
```

```
typedef enum{
    EMV_AUTHORIZATION_RESULT_ACCEPTED = 0X00,
    EMV_AUTHORIZATION_RESULT_UNABLE_TO_GO_ONLINE = 0X01,
    EMV_AUTHORIZATION_RESULT_TECHNICAL_ISSUE = 0X02,
    EMV_AUTHORIZATION_RESULT_DECLINED = 0X03,
    EMV_AUTHORIZATION_RESULT_ISSUER_REFERAL = 0X04
} EMV_AUTHORIZATION_RESULT;
```

Chapter 7

UniPay Error Code Reference

0000	No error, beginning task
0001	No response from reader
0002	Invalid response data
0003	Time out for task or CMD
0004	Wrong parameter
0005	SDK is doing MSR or ICC task
0006	SDK is doing PINPad task
0007	SDK is doing Other task
0300	Key Type(TDES) of Session Key is not same as the related Master Key.
0400	Related Key was not loaded.
0500	Key Same.
0702	PAN is Error Key.
0D00	This Key had been loaded.
0E00	Base Time was loaded.
1800	Send "Cancel Command" after send "Get Encrypted PIN" &"Get Numeric "& "Get Amount"
1900	Press "Cancel" key after send "Get Encrypted PIN" &"Get Numeric "& "Get Amount"
30FF	Security Chip is not connect
3000	Security Chip is deactivation & Device is In Removal Legally State.
3101	Security Chip is activation & Device is In Removal Legally State.
5500	No Admin DUKPT Key.
5501	Admin DUKPT Key STOP.
5502	Admin DUKPT Key KSN is Error.
5503	Get Authentication Code1 Failed.
5504	Validate Authentication Code Error.
5505	Encrypt or Decrypt data failed.
5506	Not Support the New Key Type.
5507	New Key Index is Error.
5508	Step Error.
550F	Other Error.
6000	Save or Config Failed / Or Read Config Error.
6200	No Serial Number.
6900	Invalid Command - Protocol is right, but task ID is invalid.
6A00	Unsupported Command - Protocol and task ID are right, but command is invalid.
6B00	Unknown parameter in command - Protocol task ID and command are right, but parameter is invalid.
7200	Device is suspend (MKSK suspend or press password suspend).
7300	PIN DUKPT is STOP (21 bit 1).
7400	Device is Busy.
E100	Can not enter sleep mode.
E200	File has existed.
E300	File has not existed.
E400	Open File Error.
E500	SmartCard Error.
E600	Get MSR Card data is error.
E700	Command time out.
E800	File read or write is error.
E900	Active 1850 error!
EA00	Load bootloader error.
EF00	Protocol Error- STX or ETX or check error.
EB00	Picture is not exist.
2C06	no card seated to request ATR
2D01	Card Not Supported,
2D03	Card Not Supported, wants CRC
690D	Command not supported on reader without ICC support
8100	ICC error time out on power-up
8200	invalid TS character received
8500	pps confirmation error
8600	Unsupported F, D, or combination of F and D
8700	protocol not supported EMV TD1 out of range
8800	power not at proper level
8900	ATR length too long
8B01	EMV invalid TA1 byte value
8B02	EMV TB1 required
8B03	EMV Unsupported TB1 only 00 allowed

```
8B04    EMV Card Error, invalid BWI or CWI
8B06    EMV TB2 not allowed in ATR
8B07    EMV TC2 out of range
8B08    EMV TC2 out of range
8B09    per EMV96 TA3 must be > 0xF
8B10    ICC error on power-up
8B11    EMV T=1 then TB3 required
8B12    Card Error, invalid BWI or CWI
8B13    Card Error, invalid BWI or CWI
8B17    EMV TC1/TB3 conflict*
8B20    EMV TD2 out of range must be T=1
8C00    TCK error
A304    connector has no voltage setting
A305    ICC error on power-up invalid (SBLK(IFSD) exchange
E301    ICC error after session start
FF00    Request to go online
FF01    EMV: Accept the offline transaction
FF02    EMV: Decline the offline transaction
FF03    EMV: Accept the online transaction
FF04    EMV: Decline the online transaction
FF05    EMV: Application may fallback to magstripe technology
FF06    EMV: ICC detected tah the conditions of use are not satisfied
FF07    EMV: ICC didn't accept transaction
FF08    EMV: Transaction was cancelled
FF09    EMV: Application was not selected by kernel or ICC format error or ICC missing data error
FF0A    EMV: Transaction is terminated
FF0B    EMV: Other EMV Error
FFFF    NO RESPONSE

0008    err response or data
0009    no reader attached
000A    did connection
000B    mono audio is enabled
000C    audio volume is too low
000D    task or CMD be canceled

0E00    Authorization Accepted
0E01    Unable to go online
0E02    Technical Issue
0E03    Declined
0E04    Issuer Referral transaction

0F00    Accept the online transaction
0F01    Decline the online transaction
0F02    Request to go online
0F03    Transaction is terminated
0F05    Application was not selected by kernel or ICC format error or ICC missing data error
0F07    ICC didn't accept transaction
0F0A    Application may fallback to magstripe technology
0F0C    Transaction was cancelled
0F0D    Timeout
0F0F    Other EMV Error
0F10    Accept the offline transaction
0F11    Decline the offline transaction

0F21    ICC detected tah the conditions of use are not satisfied
0F22    No app were found on card matching terminal configuration
0F23    Terminal file does not exist
0F24    CAPK file does not exist
0F25    CRL Entry does not exist
0FFE    Return code when blocking is disabled
0FFF    Return code when command is not applicable on the selected device
```

Chapter 8

EMV Tag Reference

Tag	Description
42	Issuer Identification Number (IIN)
4F	Application Identifier (ADF Name)
50	Application Label
52	Command to perform
56	Track 1 Data
57	Track 2 Equivalent Data
5A	Application Primary Account Number (PAN)
5D	Deleted (see 9D)
5F20	Cardholder Name
5F24	Application Expiration Date
5F25	Application Effective Date
5F28	Issuer Country Code
5F2A	Transaction Currency Code
5F2D	Language Preference
5F30	Service Code
5F34	Application Primary Account Number (PAN) Sequence Number (PSN)
5F36	Transaction Currency Exponent
5F3C	Transaction Reference Currency Code
5F3D	Transaction Reference Currency Exponent
5F50	Issuer URL
5F53	International Bank Account Number (IBAN)
5F54	Bank Identifier Code (BIC)
5F55	Issuer Country Code (alpha2 format)
5F56	Issuer Country Code (alpha3 format)
5F57	Account Type
61	Application Template
62	File Control Parameters (FCP) Template
6F	File Control Information (FCI) Template
70	READ RECORD Response Message Template
71	Issuer Script Template 1
72	Issuer Script Template 2
73	Directory Discretionary Template
77	Response Message Template Format 2
80	Response Message Template Format 1
81	Amount, Authorised (Binary)

Tag	Description
82	Application Interchange Profile (AIP)
83	Command Template
84	Dedicated File (DF) Name
86	Issuer Script Command
87	Application Priority Indicator
88	Short File Identifier (SFI)
89	Authorisation Code
8A	Authorisation Response Code (ARC)
8C	Card Risk Management Data Object List 1 (CDOL1)
8D	Card Risk Management Data Object List 2 (CDOL2)
8E	Cardholder Verification Method (CVM) List
8F	Certification Authority Public Key Index (PKI)
90	Issuer Public Key Certificate
91	Issuer Authentication Data
92	Issuer Public Key Remainder
93	Signed Application Data
94	Application File Locator (AFL)
95	Terminal Verification Results (TVR)
97	Transaction Certificate Data Object List (TDOL)
98	Transaction Certificate (TC) Hash Value
99	Transaction Personal Identification Number (PIN) Data
9A	Transaction Date
9B	Transaction Status Information
9C	Transaction Type
9D	Directory Definition File (DDF) Name
9F01	Acquirer Identifier
9F02	Amount, Authorised (Numeric)
9F03	Amount, Other (Numeric)
9F04	Amount, Other (Binary)
9F05	Application Discretionary Data
9F06	Application Identifier (AID) - terminal
9F07	Application Usage Control (AUC)
9F08	Application Version Number
9F09	Application Version Number
9F0B	Cardholder Name Extended
9F0D	Issuer Action Code - Default
9F0E	Issuer Action Code - Denial
9F0F	Issuer Action Code - Online
9F10	Issuer Application Data (IAD)
9F11	Issuer Code Table Index
9F12	Application Preferred Name
9F13	Last Online Application Transaction Counter (ATC) Register
9F14	Lower Consecutive Offline Limit
9F15	Merchant Category Code
9F16	Merchant Identifier
9F17	Personal Identification Number (PIN) Try Counter
9F18	Issuer Script Identifier
9F19	Deleted (see 9F49)
9F1A	Terminal Country Code

Tag	Description
9F1B	Terminal Floor Limit
9F1C	Terminal Identification
9F1D	Terminal Risk Management Data
9F1E	Interface Device (IFD) Serial Number
9F1F	Track 1 Discretionary Data
9F20	Track 2 Discretionary Data
9F21	Transaction Time
9F22	Certification Authority Public Key Index (PKI)
9F23	Upper Consecutive Offline Limit
9F26	Application Cryptogram (AC)
9F27	Cryptogram Information Data (CID)
9F29	Extended Selection
9F2A	Kernel Identifier
9F2D	Integrated Circuit Card (ICC) PIN Encipherment Public Key Certificate
9F2E	Integrated Circuit Card (ICC) PIN Encipherment Public Key Exponent
9F2F	Integrated Circuit Card (ICC) PIN Encipherment Public Key Remainder
9F32	Issuer Public Key Exponent
9F33	Terminal Capabilities
9F34	Cardholder Verification Method (CVM) Results
9F35	Terminal Type
9F36	Application Transaction Counter (ATC)
9F37	Unpredictable Number (UN)
9F37	Unpredictable Number (UN) (Reader/Terminal)
9F38	Processing Options Data Object List (PDOL)
9F39	Point-of-Service (POS) Entry Mode
9F3A	Amount, Reference Currency
9F3B	Application Reference Currency
9F3C	Transaction Reference Currency Code
9F3D	Transaction Reference Currency Exponent
9F40	Additional Terminal Capabilities
9F41	Transaction Sequence Counter
9F42	Application Currency Code
9F43	Application Reference Currency Exponent
9F44	Application Currency Exponent
9F45	Data Authentication Code
9F46	Integrated Circuit Card (ICC) Public Key Certificate
9F46	Application Public Key Certificate
9F47	Integrated Circuit Card (ICC) Public Key Exponent
9F47	Application Public Key Exponent
9F48	Integrated Circuit Card (ICC) Public Key Remainder
9F48	Application Public Key Remainder
9F49	Dynamic Data Authentication Data Object List (DDOL)
9F4A	Static Data Authentication Tag List (SDA)
9F4B	Signed Dynamic Application Data (SDAD)
9F4C	ICC Dynamic Number
9F4D	Log Entry
9F4E	Merchant Name and Location
9F4F	Log Format
9F50	Offline Accumulator Balance

Tag	Description
9F50	Cardholder Verification Status
9F51	Application Currency Code
9F51	DRDOL
9F52	Application Default Action (ADA)
9F52	Terminal Compatibility Indicator
9F53	Consecutive Transaction Counter International Limit (CTCIL)
9F53	Transaction Category Code
9F53	Terminal Interchange Profile (dynamic)
9F54	Cumulative Total Transaction Amount Limit (CTTAL)
9F54	DS ODS Card
9F55	Geographic Indicator
9F56	Issuer Authentication Indicator
9F57	Issuer Country Code
9F58	Consecutive Transaction Counter Limit (CTCL)
9F59	Consecutive Transaction Counter Upper Limit (CTCUL)
9F5A	Application Program Identifier (Program ID)
9F5B	Issuer Script Results
9F5B	DSDOL
9F5C	Cumulative Total Transaction Amount Upper Limit (CTTAUL)
9F5C	DS Requested Operator ID
9F5C	Magstripe Data Object List (MDOL)
9F5D	Available Offline Spending Amount (AOSA)
9F5D	Application Capabilities Information (ACI)
9F5E	Consecutive Transaction International Upper Limit (CTIUL)
9F5E	DS ID
9F5F	DS Slot Availability
9F5F	Offline Balance
9F60	CVC3 (Track1)
9F60	Issuer Update Parameter
9F60	P3 Generated 3DES KEYS
9F61	CVC3 (Track2)
9F62	PCVC3 (Track1)
9F62	Encrypted PIN - ISO 95641 Format 0 (Thales P3 Format 01)
9F63	Offline Counter Initial Value
9F63	PUNATC (Track1)
9F64	NATC (Track1)
9F65	PCVC3 (Track2)
9F66	Terminal Transaction Qualifiers (TTQ)
9F66	PUNATC (Track2)
9F67	MSD Offset
9F67	NATC (Track2)
9F68	Card Additional Processes
9F69	Card Authentication Related Data
9F69	UDOL
9F6A	Unpredictable Number (Numeric)
9F6B	Card CVM Limit
9F6B	Track 2 Data
9F6C	Card Transaction Qualifiers (CTQ)
9F6D	VLP Reset Threshold
9F6D	Mag-stripe Application Version Number (Reader)

Tag	Description
9F6D	Kernel 4 Reader Capabilities
9F6E	Third Party Data
9F6E	Form Factor Indicator (FFI)
9F6E	Terminal Transaction Capabilities
9F6F	DS Slot Management Control
9F70	Protected Data Envelope 1
9F70	Card Interface Capabilities
9F71	Protected Data Envelope 2
9F71	Mobile CVM Results
9F72	Protected Data Envelope 3
9F72	Consecutive Transaction Limit (International—Country)
9F73	Protected Data Envelope 4
9F73	Currency Conversion Parameters
9F74	Protected Data Envelope 5
9F74	VLP Issuer Authorisation Code
9F75	Unprotected Data Envelope 1
9F75	Cumulative Total Transaction Amount Limit-Dual Currency
9F76	Unprotected Data Envelope 2
9F76	Secondary Application Currency Code
9F77	Unprotected Data Envelope 3
9F78	Unprotected Data Envelope 4
9F79	Unprotected Data Envelope 5
9F77	VLP Funds Limit
9F78	VLP Single Transaction Limit
9F79	VLP Available Funds
9F7A	VLP Terminal Support Indicator
9F7B	VLP Terminal Transaction Limit
9F7C	Customer Exclusive Data (CED)
9F7C	Merchant Custom Data
9F7D	DS Summary 1
9F7D	VISA Applet Data
9F7E	Mobile Support Indicator
9F7E	Application life cycle data (8 first bytes)
9F7F	DS Unpredictable Number
9F7F	Card Production Life Cycle (CPLC) Data
A5	File Control Information (FCI) Proprietary Template
BF0C	File Control Information (FCI) Issuer Discretionary Data
BF50	Visa Fleet - CDO
BF60	Integrated Data Storage Record Update Template
C3	Card issuer action code -decline
C4	Card issuer action code -default
C5	Card issuer action code online
C6	PIN Try Limit
C7	CDOL 1 Related Data Length
C8	Card risk management country code
C9	Card risk management currency code
CA	Lower cummulative offline transaction amount
CB	Upper cumulative offline transaction amount
CD	Card Issuer Action Code (PayPass) – Default

Tag	Description
CE	Card Issuer Action Code (PayPass) – Online
CF	Card Issuer Action Code (PayPass) – Decline
D1	Currency conversion table
D2	Integrated Data Storage Directory (IDSD)
D3	Additional check table
D5	Application Control
D6	Default ARPC response code
D7	Application Control (PayPass)
D8	AIP (PayPass)
D9	AFL (PayPass)
DA	Static CVC3-TRACK1
DB	Static CVC3-TRACK2
DC	IVCVC3-TRACK1
DD	IVCVC3-TRACK2
DF01	Encrypted PIN Block in Tag 9F62 – ISO 95641 Format 0
DF02	PEK Version Number
DF03	PIN Try Limit
DF04	PIN Try Counter (VSDC Application)
DF05	AIP - For VISA Contactless
DF06	Products permitted
DF07	Offline checks mandated
DF08	UDKmac
DF09	UDKenc
DF0B	Retries Permitted Limit
DF0C	Script Message Update
DF0D	Fleet Issuer Action Code - Default
DF0E	Fleet Issuer Action Code - Denial
DF0F	Fleet Issuer Action Code - Online
DF12	Vehicle Registration Number
DF13	DDA Public Modulus
DF14	Driver Name
DF15	Driver ID
DF16	Max Fill Volume
DF17	DDA Public Modulus Length
DF18	Mileage
DF20	Issuer Proprietary Bitmap (IPB)
DF21	Internet Authentication Flag (IAF)
DF22	Encrypted PEK - RFU
DF23	PEK Key Check Value - RFU
DF24	MDK - Key derivation Index
DF25	VISA DPA – MDK - Key derivation Index
DF26	Encrypted PIN Block – ISO 9564-1 Format 1 PIN Block (Thales P3 Format 05)
DF40	qVSDC AIP
DF41	VSDC AIP
DF42	UDKac
DF43	UDKmac
DF44	UDKenc
DF47	UDKcvc
DF48	UDKac KCV
DF49	UDKmac KCV
DF4A	UDKenc KCV

Tag	Description
DF4B	UDKcvc KCV
DF4B	POS Cardholder Interaction Information
DF51	Grand Parent AC
DF52	Parent AC
DF53	Grand Parent MAC
DF54	Parent MAC
DF55	Grand Parent ENC
DF56	Parent ENC/Terminal Action Code - Default
DF57	Terminal Action Code - Decline
DF60	DS Input (Card)
DF60	DDA Component P
DF61	DDA Component Q
DF61	DS Digest H
DF62	DS ODS Info
DF62	DDA Component D1
DF63	DDA Component D2
DF63	DS ODS Term
DF64	DDA Component Q Minus 1 Mod P
DF65	DDA Private Exponent
DF6B	Paypass Contactless
DF79	Dynamic Data Authentication Keys
DF8101	DS Summary 2
DF8102	DS Summary 3
DF8104	Balance Read Before Gen AC
DF8105	Balance Read After Gen AC
DF8106	Data Needed
DF8107	CDOL1 Related Data
DF8108	DS AC Type
DF8109	DS Input (Term)
DF810A	DS ODS Info For Reader
DF810B	DS Summary Status
DF810C	Kernel ID
DF810D	DSVN Term
DF810E	Post-Gen AC Put Data Status
DF810F	Pre-Gen AC Put Data Status
DF8110	Proceed To First Write Flag
DF8111	PDOL Related Data
DF8112	Tags To Read
DF8113	DRDOL Related Data
DF8114	Reference Control Parameter
DF8115	Error Indication
DF8116	User Interface Request Data
DF8117	Card Data Input Capability
DF8118	CVM Capability – CVM Required
DF8119	CVM Capability – No CVM Required
DF811A	Default UDOL
DF811B	Kernel Configuration
DF811C	Max Lifetime of Torn Transaction Log Record
DF811D	Max Number of Torn Transaction Log Records
DF811E	Mag-stripe CVM Capability – CVM Required

Tag	Description
DF811F	Security Capability
DF8120	Terminal Action Code – Default
DF8121	Terminal Action Code – Denial
DF8122	Terminal Action Code – Online
DF8123	Reader Contactless Floor Limit
DF8124	Reader Contactless Transaction Limit (No On-device CVM)
DF8125	Reader Contactless Transaction Limit (On-device CVM)
DF8126	Reader CVM Required Limit
DF8127	Time Out Value
DF8128	IDS Status
DF8129	Outcome Parameter Set
DF812A	DD Card (Track1)
DF812B	DD Card (Track2)
DF812C	Mag-stripe CVM Capability – No CVM Required
DF812D	Message Hold Time
DF8130	Hold Time Value
DF8131	Phone Message Table
FF60	Visa International
FF62	Visa Magnetic Stripe
FF63	Visa Quick VSDC
FF8101	Torn Record
FF8102	Tags To Write Before Gen AC
FF8103	Tags To Write After Gen AC
FF8104	Data To Send
FF8105	Data Record
FF8106	Discretionary Data

Chapter 9

Hierarchical Index

9.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 10

Data Structure Index

10.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 11

Data Structure Documentation

11.1 ICCReaderStatus Struct Reference

`#include <IDTCommon.h>`

Data Fields

- bool iccPower

  Determines if ICC has been powered up.
- bool cardSeated

  Determines if card is inserted.
- bool latchClosed

  Determines if Card Latch is engaged. If device does not have a latch, value is always FALSE.
- bool cardPresent

  If device has a latch, determines if the card is present in device. If the device does not have a latch, value is always FALSE.
- bool magneticDataPresent

  True = Magnetic data present, False = No Magnetic Data.

11.1.1 Detailed Description

Structure used to return response from IDT_BTPay::icc_getICCReaderStatus() and IDT_UniPay::icc_getICC↩ ReaderStatus()

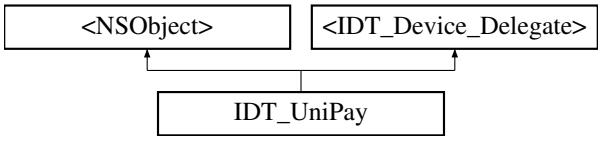
The documentation for this struct was generated from the following file:

- SourceMac/IDTCommon.h

11.2 IDT_UniPay Class Reference

`#import <IDT_UniPay.h>`

Inheritance diagram for IDT_UniPay:



Instance Methods

- (RETURN_CODE) - config_getModelNumber:
- (RETURN_CODE) - config_getSerialNumber:
- (BOOL) - config_setCmdTimeOutDuration:
- (RETURN_CODE) - config_setSerialNumber:
- (RETURN_CODE) - device_cancelConnectToAudioReader
- (RETURN_CODE) - device_connectToAudioReader
- (RETURN_CODE) - device_getBatteryVoltage:
- (RETURN_CODE) - device_getFirmwareVersion:
- (RETURN_CODE) - device_getKSN:ksn:
- (RETURN_CODE) - device_getLevelAndBaud:
- (NSString ∗) - device_getResponseCodeString:
- (BOOL) - device_isAudioReaderConnected
- (bool) - device_isConnected:
- (RETURN_CODE) - device_rebootDevice
- (RETURN_CODE) - device_sendDataCommand:calcLRC:response:
- (RETURN_CODE) - device_setAudioVolume:
- (RETURN_CODE) - device_startRKI
- (RETURN_CODE) - icc_exchangeAPDU:encrypted:response:
- (RETURN_CODE) - icc_exchangeEncryptedAPDU:response:
- (RETURN_CODE) - icc_exchangeMultiAPDU:response:
- (RETURN_CODE) - icc_getAPDU_KSN:
- (RETURN_CODE) - icc_getExpiryDateOption:
- (RETURN_CODE) - icc_getICCReaderStatus:
- (RETURN_CODE) - icc_getKeyFormatForICCDUKPT:
- (RETURN_CODE) - icc_getKeyTypeForICCDUKPT:
- (RETURN_CODE) - icc_powerOffICC:
- (RETURN_CODE) - icc_powerOnICC:
- (RETURN_CODE) - icc_setICCNotification:
- (RETURN_CODE) - icc_setKeyFormatForICCDUKPT:
- (RETURN_CODE) - icc_setKeyTypeForICCDUKPT:
- (RETURN_CODE) - icc_loadDUKPTKey:ksn:initialKey:
- (RETURN_CODE) - msr_cancelMSRSwipe
- (RETURN_CODE) - msr_getClearPANID:
- (RETURN_CODE) - msr_getExpirationMask:
- (RETURN_CODE) - msr_getSwipeEncryption:
- (RETURN_CODE) - msr_getSwipeForcedEncryptionOption:
- (RETURN_CODE) - msr_getSwipeMaskOption:
- (RETURN_CODE) - msr_setClearPANID:
- (RETURN_CODE) - msr_setExpirationMask:
- (RETURN_CODE) - msr_setSwipeEncryption:
- (RETURN_CODE) - msr_setSwipeForcedEncryptionOption:track2:track3:track3card0:
- (RETURN_CODE) - msr_setSwipeMaskOption:track2:track3:
- (RETURN_CODE) - msr_startMSRSwipe:
- (bool) - isConnected
- (void) - attemptConnect

Class Methods

- (NSString ∗) + SDK_version
- (IDT_UniPay ∗) + sharedController

Properties

- id< IDT_UniPay_Delegate > delegate

11.2.1 Detailed Description

Class to drive the IDT_UniPay device

11.2.2 Method Documentation

11.2.2.1 - (void) attemptConnect

Attempt connection

Requests a connection attempt. Internal use only.

11.2.2.2 - (RETURN_CODE) config_getModelNumber: (NSString **) *response*

Polls device for Model Number

Parameters

response	Returns Model Number

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.3 - (RETURN_CODE) config_getSerialNumber: (NSString **) *response*

Polls device for Serial Number

Parameters

response	Returns Serial Number

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.4 - (BOOL) config_setCmdTimeOutDuration: (float) *nSecond*

Command Acknowledgement Timout

Sets the amount of seconds to wait for an {ACK} to a command before a timeout. Responses should normally be received under one second. Default is 3 seconds

Parameters

nSecond	Timout value. Valid range .1 - 60 seconds

Returns

Success flag. Determines if value was set and in range.

11.2.2.5 - (RETURN_CODE) config_setSerialNumber: (NSString *) *strSN*

Set Serial Number

Set device's serial number and Bluetooth name, then reboots device. Bluetooth name will be set as IDT_UniPay + Space + Serial number

Parameters

strSN	Device serial number

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.6 - (RETURN_CODE) device_cancelConnectToAudioReader

Cancel Connect To Audio Reader

Returns

RETURN_CODE

Cancels a connection attempt to an IDTech MSR device connected via the audio port.

11.2.2.7 - (RETURN_CODE) device_connectToAudioReader

Connect To Audio Reader

Returns

RETURN_CODE

Attemps to recognize and connect to an IDTech MSR device connected via the audio port.

11.2.2.8 - (RETURN_CODE) device_getBatteryVoltage: (NSString **) *response*

Polls device for Battery Voltage

Parameters

response	Returns Battery Voltage as 4-chararacter string * 100. Example: "0186" = 1.86v. "1172" = 11.72v.

Returns

RETURN_CODE:

- 0x0000: Success: no error - RETURN_CODE_DO_SUCCESS
- 0x0001: Disconnect: no response from reader - RETURN_CODE_ERR_DISCONNECT

- 0x0002: Invalid Response: invalid response data - RETURN_CODE_ERR_CMD_RESPONSE
- 0x0003: Timeout: time out for task or CMD - RETURN_CODE_ERR_TIMEDOUT
- 0x0004: Invalid Parameter: wrong parameter - RETURN_CODE_ERR_INVALID_PARAMETER
- 0x0005: MSR Busy: SDK is doing MSR or ICC task - RETURN_CODE_SDK_BUSY_MSR
- 0x0006: PINPad Busy: SDK is doing PINPad task - RETURN_CODE_SDK_BUSY_PINPAD
- 0x0007: Unknown: Unknown error - RETURN_CODE_ERR_OTHER
- 0x0100 through 0xFFFF refer to IDT_Device::getResponseCodeString:()

11.2.2.9 - (RETURN_CODE) device_getFirmwareVersion: (NSString **) *response*

Polls device for Firmware Version

Parameters

response	Response returned of Firmware Version

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.10 - (RETURN_CODE) device_getKSN: (int) *keySlot* ksn:(NSData **) *ksn*

Get Account KSN

Retrieves the KSN for a key slot

Parameters

keySlot	0x02=MSR DUKPT Key, 0x0C = Admin DUKPT Key, 0x22 = ICC DUKPT Key
ksn	Returns the Account DUKPT KSN

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE

- If Key was not loaded, unit should respond error code 0x0400.
- If Key is end of useful life, unit should respond error code 0x7300

11.2.2.11 - (RETURN_CODE) device_getLevelAndBaud: (NSString **) *response*

Get Level and Baude

Parameters

response	The Baud Rate and Audio Level.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

- 0x0100 through 0xFFFF refer to IDT_Device::getResponseCodeString:()

11.2.2.12 - (NSString ∗) device_getResponseCodeString: (int) *errorCode*

Get Response Code String

Interpret a IDT_UniPay response code and return string description.

Parameters

errorCode	Error code, range 0x0000 - 0xFFFF, example 0x0300

Returns

Verbose error description

HEX VALUE	Description
0x0000	No error, beginning task
0x0001	No response from reader
0x0002	Invalid response data
0x0003	Time out for task or CMD
0x0004	Wrong parameter
0x0005	SDK is doing MSR or ICC task
0x0006	SDK is doing PINPad task
0x0007	SDK is doing Other task
0x0300	Key Type(TDES) of Session Key is not same as the related Master Key.
0x0400	Related Key was not loaded.
0x0500	Key Same.
0x0702	PAN is Error Key.
0x0D00	This Key had been loaded.
0x0E00	Base Time was loaded.
0x1800	Send “Cancel Command” after send “Get Encrypted PIN” &”Get Numeric “& “Get Amount”
0x1900	Press “Cancel” key after send “Get Encrypted PIN” &”Get Numeric “& “Get Amount”
0x30FF	Security Chip is not connect
0x3000	Security Chip is deactivation & Device is In Removal Legally State.
0x3101	Security Chip is activation & Device is In Removal Legally State.
0x5500	No Admin DUKPT Key.
0x5501	Admin DUKPT Key STOP.
0x5502	Admin DUKPT Key KSN is Error.
0x5503	Get Authentication Code1 Failed.
0x5504	Validate Authentication Code Error.
0x5505	Encrypt or Decrypt data failed.
0x5506	Not Support the New Key Type.
0x5507	New Key Index is Error.
0x5508	Step Error.
0x550F	Other Error.
0x6000	Save or Config Failed / Or Read Config Error.

HEX VALUE	Description
0x6200	No Serial Number.
0x6900	Invalid Command - Protocol is right, but task ID is invalid.
0x6A00	Unsupported Command - Protocol and task ID are right, but command is invalid.
0x6B00	Unknown parameter in command - Protocol task ID and command are right, but parameter is invalid.
0x7200	Device is suspend (MKSK suspend or press password suspend).
0x7300	PIN DUKPT is STOP (21 bit 1).
0x7400	Device is Busy.
0xE100	Can not enter sleep mode.
0xE200	File has existed.
0xE300	File has not existed.
0xE400	Open File Error.
0xE500	SmartCard Error.
0xE600	Get MSR Card data is error.
0xE700	Command time out.
0xE800	File read or write is error.
0xE900	Active 1850 error!
0xEA00	Load bootloader error.
0xEF00	Protocol Error- STX or ETX or check error.
0xEB00	Picture is not exist.
0x2C06	no card seated to request ATR
0x2D01	Card Not Supported,
0x2D03	Card Not Supported, wants CRC
0x690D	Command not supported on reader without ICC support
0x8100	ICC error time out on power-up
0x8200	invalid TS character received
0x8500	pps confirmation error
0x8600	Unsupported F, D, or combination of F and D
0x8700	protocol not supported EMV TD1 out of range
0x8800	power not at proper level
0x8900	ATR length too long
0x8B01	EMV invalid TA1 byte value
0x8B02	EMV TB1 required
0x8B03	EMV Unsupported TB1 only 00 allowed
0x8B04	EMV Card Error, invalid BWI or CWI
0x8B06	EMV TB2 not allowed in ATR
0x8B07	EMV TC2 out of range
0x8B08	EMV TC2 out of range
0x8B09	per EMV96 TA3 must be $>$ 0xF
0x8B10	ICC error on power-up
0x8B11	EMV T=1 then TB3 required
0x8B12	Card Error, invalid BWI or CWI
0x8B13	Card Error, invalid BWI or CWI
0x8B17	EMV TC1/TB3 conflict$*$
0x8B20	EMV TD2 out of range must be T=1
0x8C00	TCK error
0xA304	connector has no voltage setting
0xA305	ICC error on power-up invalid (SBLK(IFSD) exchange
0xE301	ICC error after session star
0xFF00	EMV: Request to go online

HEX VALUE	Description
0xFF01	EMV: Accept the offline transaction
0xFF02	EMV: Decline the offline transaction
0xFF03	EMV: Accept the online transaction
0xFF04	EMV: Decline the online transaction
0xFF05	EMV: Application may fallback to magstripe technology
0xFF06	EMV: ICC detected that the conditions of use are not satisfied
0xFF07	EMV: ICC didn't accept transaction
0xFF08	EMV: Transaction was cancelled
0xFF09	EMV: Application was not selected by kernel or ICC format error or ICC missing data error
0xFF0A	EMV: Transaction is terminated
0xFF0B	EMV: Other EMV Error

11.2.2.13 - (BOOL) device_isAudioReaderConnected

Is Audio Reader Connected

Returns value on device connection status when device is an audio-type connected to headphone plug.

Returns

BOOL True = Connected, False = Disconnected

11.2.2.14 - (bool) device_isConnected: (IDT_DEVICE_Types) *device*

Is Device Connected

Returns the connection status of the requested device

Parameters

device	Check connectivity of device type

```
typedef enum{
IDT_DEVICE_UniPay_IOS = 3,
IDT_DEVICE_UniPay_OSX_USB = 4
}IDT_DEVICE_Types;
```

11.2.2.15 - (RETURN_CODE) device_rebootDevice

Reboot Device

Executes a command to restart the device.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.16 - (RETURN_CODE) device_sendDataCommand: (NSData ∗) *cmd* calcLRC:(BOOL) *lrc* response:(NSData ∗∗) *response*

Send a NSData object to device

Sends a command represented by the provide NSData object to the device through the accessory protocol.

Parameters

cmd	NSData representation of command to execute
lrc	If `TRUE`, this will wrap command with start/length/lrc/sum/end: '{STX}{Len_Low}{Len_High} data {CheckLRC} {CheckSUM} {ETX}'
response	Response data

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:)

11.2.2.17 - (RETURN_CODE) device_setAudioVolume: (float) *val*

Set Volume To Audio Reader

Set the iPhone's volume for command communication with audio-based readers. The the range of iPhone's volume is from 0.1 to 1.0.

Parameters

val	Volume level from 0.1 to 1.0

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.18 - (RETURN_CODE) device_startRKI

Start Remote Key Injection

Attempts to perform a Remote Key Injection with IDTech's RKI servers.

Returns

RETURN_CODE:

- 0x0000: Success: no error - RETURN_CODE_DO_SUCCESS
- 0x0001: Disconnect: no response from reader - RETURN_CODE_ERR_DISCONNECT
- 0x0002: Invalid Response: invalid response data - RETURN_CODE_ERR_CMD_RESPONSE
- 0x0003: Timeout: time out for task or CMD - RETURN_CODE_ERR_TIMEDOUT
- 0x0004: Invalid Parameter: wrong parameter - RETURN_CODE_ERR_INVALID_PARAMETER
- 0x0005: MSR Busy: SDK is doing MSR or ICC task - RETURN_CODE_SDK_BUSY_MSR
- 0x0006: PINPad Busy: SDK is doing PINPad task - RETURN_CODE_SDK_BUSY_PINPAD
- 0x0100 through 0xFFFF refer to IDT_Device::getResponseCodeString:()

11.2.2.19 - (RETURN_CODE) icc_exchangeAPDU: (NSData ∗) *dataAPDU* encrypted:(BOOL) *encrypted* response:(APDUResponse ∗∗) *response*

Exchange APDU

Sends an APDU packet to the ICC. If successful, response is returned in APDUResult class instance in response parameter.

Parameters

dataAPDU	APDU data packet
encrypted	Send data encrypted for special case
response	Unencrypted/encrypted parsed APDU response

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:)

11.2.2.20 - (RETURN_CODE) icc_exchangeEncryptedAPDU: (NSData ∗) *dataAPDU* response:(APDUResponse ∗∗) *response*

Exchange Encrypted APDU

- UniPay

  Sends an encrypted APDU packet to the ICC. If successful, response is returned in APDUResult class instance in response parameter.

Parameters

dataAPDU	APDU data packet
response	encrypted parsed APDU response

Returns

RETURN_CODE:

- 0x0000: Success: no error - RETURN_CODE_DO_SUCCESS
- 0x0001: Disconnect: no response from reader - RETURN_CODE_ERR_DISCONNECT
- 0x0002: Invalid Response: invalid response data - RETURN_CODE_ERR_CMD_RESPONSE
- 0x0003: Timeout: time out for task or CMD - RETURN_CODE_ERR_TIMEDOUT
- 0x0004: Invalid Parameter: wrong parameter - RETURN_CODE_ERR_INVALID_PARAMETER
- 0x0005: MSR Busy: SDK is doing MSR or ICC task - RETURN_CODE_SDK_BUSY_MSR
- 0x0006: PINPad Busy: SDK is doing PINPad task - RETURN_CODE_SDK_BUSY_PINPAD
- 0x0007: Unknown: Unknown error - RETURN_CODE_ERR_OTHER
- 0x0100 through 0xFFFF refer to IDT_Device::getResponseCodeString:()

11.2.2.21 - (RETURN_CODE) icc_exchangeMultiAPDU: (NSArray ∗) *dataAPDU* response:(NSData ∗∗) *response*

Exchange Multi APDU

Sends multiple APDU commands within on command

Parameters

dataAPDU	An array of NSData APDU commands
response	The combined response of the multiple APDU commands

Returns

RETURN_CODE:

- 0x0000: Success: no error - RETURN_CODE_DO_SUCCESS
- 0x0001: Disconnect: no response from reader - RETURN_CODE_ERR_DISCONNECT
- 0x0002: Invalid Response: invalid response data - RETURN_CODE_ERR_CMD_RESPONSE
- 0x0003: Timeout: time out for task or CMD - RETURN_CODE_ERR_TIMEDOUT
- 0x0004: Invalid Parameter: wrong parameter - RETURN_CODE_ERR_INVALID_PARAMETER
- 0x0005: MSR Busy: SDK is doing MSR or ICC task - RETURN_CODE_SDK_BUSY_MSR
- 0x0006: PINPad Busy: SDK is doing PINPad task - RETURN_CODE_SDK_BUSY_PINPAD
- 0x0007: Unknown: Unknown error - RETURN_CODE_ERR_OTHER
- 0x0100 through 0xFFFF refer to IDT_Device::getResponseCodeString:()

11.2.2.22 - (RETURN_CODE) icc_getAPDU_KSN: (NSData **) *ksn*

Get APDU KSN

Retrieves the KSN used in ICC Encypted APDU usage

Parameters

ksn	Returns the encrypted APDU packet KSN

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE

11.2.2.23 - (RETURN_CODE) icc_getExpiryDateOption: (NSString **) *response*

Get Expiry Date Option UniPay

Executes a command to get the Expiry Date Option.

Parameters

response	Expiry Option.. • "0" Output masked for Tag 57 and only output encrypted for Tag 5F24 • "1" Output plaintext

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.24 - (RETURN_CODE) icc_getICCReaderStatus: (ICCReaderStatus **) *readerStatus*

Get Reader Status

Returns the reader status

Parameters

readerStatus	Pointer that will return with the ICCReaderStatus results.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

```
ICCReaderStatus readerStatus;
RETURN_CODE rt = [[IDT_UniPay sharedController] icc_getICCReaderStatus:&
      readerStatus];
if(RETURN_CODE_DO_SUCCESS != rt){
LOGI(@"Fail");
}
else{
NSString *sta;
if(readerStatus.iccPower)
sta =@"[ICC Powered]";
else
sta = @"[ICC Power not Ready]";
if(readerStatus.cardSeated)
sta =[NSString stringWithFormat:@"%@,[Card Seated]", sta];
else
sta =[NSString stringWithFormat:@"%@,[Card not Seated]", sta];

LOGI(@"Card Status = %@",sta);
}
```

11.2.2.25 - (RETURN_CODE) icc_getKeyFormatForICCDUKPT: (NSString **) *response*

Get Key Format For ICC DUKPT

Specifies how data will be encrypted with Data Key or PIN key (if DUKPT key loaded)

Parameters

response	Response returned from method: • 'TDES': Encrypted card data with TDES if DUKPT Key had been loaded.(default) • 'AES': Encrypted card data with AES if DUKPT Key had been loaded. • 'NONE': No Encryption.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.26 - (RETURN_CODE) icc_getKeyTypeForICCDUKPT: (NSString **) *response*

Get Key Type for ICC DUKPT

Specifies the key type used for ICC DUKPT encryption

Parameters

response	Response returned from method: • 'DATA': Encrypted card data with Data Key DUKPT Key had been loaded.(default) • 'PIN': Encrypted card data with PIN Key if DUKPT Key had been loaded.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.27 - (RETURN_CODE) icc_loadDUKPTKey: (DUKPT_KEY_Type) *type* ksn:(NSString ∗) *hexKSN* initialKey:(NSString ∗) *hexInitKey*

Loads the ICC DUKPT Key

Sets the key the data will be encrypted with

Parameters

type	Key Type • DUKPT_KEY_MSR = 0x00, • DUKPT_KEY_ICC = 0x01, • DUKPT_KEY_Admin = 0x10, • DUKPT_KEY_Paireing_PinPad = 0x20,
hexKSN	Key Serial Number as a Hex string
hexInitKey	Initial key as a Hex string

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:)

11.2.2.28 - (RETURN_CODE) icc_powerOffICC: (NSString ∗∗) *error*

Power Off ICC

Powers down the ICC

Parameters

error	Returns the error, if any

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

If Success, empty If Failure, ASCII encoded data of error string

11.2.2.29 - (RETURN_CODE) icc_powerOnICC: (NSData **) *response*

Power On ICC

Power up the currently selected microprocessor card in the ICC reader

Parameters

response	Response returned. If Success, ATR String. If Failure, ASCII encoded data of error string

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.30 - (RETURN_CODE) icc_setICCNotification: (BOOL) *turnON*

Set ICC Notifications

Determins if card insert/remove events are captured and sent to delegate UniPay_EventFunctionICC

Parameters

turnON	TRUE = monitor ICC card events, FALSE = ignore ICC card events. Default value is FALSE/OFF.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.31 - (RETURN_CODE) icc_setKeyFormatForICCDUKPT: (int) *encryption*

Set Key Format for ICC DUKPT

Sets how data will be encrypted, with either TDES or AES (if DUKPT key loaded)

Parameters

encryption	Encryption Type • 00: Encrypt with TDES • 01: Encrypt with AES

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.32 - (RETURN_CODE) icc_setKeyTypeForICCDUKPT: (int) *encryption*

Set Key Type for ICC DUKPT Key

Sets which key the data will be encrypted with, with either Data Key or PIN key (if DUKPT key loaded)

Parameters

encryption	Encryption Type • 00: Encrypt with Data Key • 01: Encrypt with PIN Key

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:)

11.2.2.33 - (bool) isConnected

Device Connected

Returns

isConnected Boolean indicated if UniPay is connected

11.2.2.34 - (RETURN_CODE) msr_cancelMSRSwipe

Disable MSR Swipe

Cancels MSR swipe request.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.35 - (RETURN_CODE) msr_getClearPANID: (NSString ∗∗) *response*

Get Clear PAN Digits

Returns the number of digits that begin the PAN that will be in the clear

Parameters

response	Number of digits in clear. Values are ASCII '0' - '6':

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.36 - (RETURN_CODE) msr_getExpirationMask: (NSString ∗∗) *response*

Get Expiration Masking

Get the flag that determines if to mask the expiration date

Parameters

response	'0' = masked, '1' = not-masked

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.37 - (RETURN_CODE) msr_getSwipeEncryption: (NSString **) *response*

Get Swipe Data Encryption

Returns the encryption used for sweip data

Parameters

response	'TDES', 'AES', 'NONE'

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.38 - (RETURN_CODE) msr_getSwipeForcedEncryptionOption: (NSString **) *response*

Get Swipe Data Encryption

Gets the swipe force encryption options

Parameters

response	A string with for flags separated by PIPE character f1|f2|f3|f4, example "1|0|0|1" where: • f1 = Track 1 Force Encrypt • f2 = Track 2 Force Encrypt • f3 = Track 3 Force Encrypt • f4 = Track 3 Force Encrypt when card type is 0

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.39 - (RETURN_CODE) msr_getSwipeMaskOption: (NSString **) *response*

Get Swipe Mask Option

Gets the swipe mask/clear data sending option

Parameters

response	A string with for flags separated by PIPE character f1|f2|f3, example "1|0|0" where: • f1 = Track 1 Mask Allowed • f2 = Track 2 Mask Allowed • f3 = Track 3 Mask Allowed

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.40 - (RETURN_CODE) msr_setClearPANID: (int) *digits*

Set Clear PAN Digits

Sets the amount of digits shown in the clear (not masked) at the beginning of the returned PAN value

Parameters

digits	Number of digits to show in clear. Range 0-6.

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.41 - (RETURN_CODE) msr_setExpirationMask: (BOOL) *masked*

Set Expiration Masking

Sets the flag to mask the expiration date

Parameters

masked	TRUE = mask expiration

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.42 - (RETURN_CODE) msr_setSwipeEncryption: (int) *encryption*

Set Swipe Data Encryption

Sets the swipe encryption method

Parameters

encryption	1 = TDES, 2 = AES

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.43 - (RETURN_CODE) msr_setSwipeForcedEncryptionOption: (BOOL) *track1* track2:(BOOL) *track2* track3:(BOOL) *track3* track3card0:(BOOL) *track3card0*

Set Swipe Force Encryption

Sets the swipe force encryption options

Parameters

track1	Force encrypt track 1
track2	Force encrypt track 2
track3	Force encrypt track 3
track3card0	Force encrypt track 3 when card type is 0

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.44 - (RETURN_CODE) msr_setSwipeMaskOption: (BOOL) *track1* track2:(BOOL) *track2* track3:(BOOL) *track3*

Set Swipe Mask Option

Sets the swipe mask/clear data sending option

Parameters

track1	Mask track 1 allowed
track2	Mask track 2 allowed
track3	Mask track 3 allowed

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.45 - (RETURN_CODE) msr_startMSRSwipe: (int) *track*

Enable MSR Swipe

Enables MSR, waiting for swipe to occur. Allows track selection. Returns IDTMSRData instance to deviceDelegate↩ ::swipeMSRData:()

Parameters

track	Track Selection Option

Track Selection Option	Val
Any Track	0
Track 1 Only	1
Track 2 Only	2
Track 1 & Track 2	3
Track 3 Only	4
Track 1 & Track 3	5
Track 2 & Track 3	6
All three Tracks	7

Returns

RETURN_CODE: Return codes listed as typedef enum in IDTCommon:RETURN_CODE. Values can be parsed with device_getResponseCodeString:

11.2.2.46 + (NSString∗) SDK_version

SDK Version

- All Devices

Returns the current version of IDTech.framework

Returns

Framework version

11.2.2.47 + (IDT_UniPay∗) sharedController

Singleton Instance

- All Devices

Establishes an singleton instance of IDT_UniPay class.

Returns

Instance of IDT_UniPay

11.2.3 Property Documentation

11.2.3.1 - (id< IDT_UniPay_Delegate >) delegate [read], [write], [atomic], [strong]

- Reference to IDT_UniPay_Delegate.

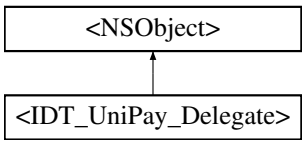
The documentation for this class was generated from the following file:

- SourceMac/IDT_UniPay.h

11.3 <IDT_UniPay_Delegate> Protocol Reference

`#import <IDT_UniPay.h>`

Inheritance diagram for <IDT_UniPay_Delegate>:



Instance Methods

- (void) - deviceConnected

  Fires when device connects. If a connection is established before the delegate is established (no delegate to send initial connection notification to), this method will fire upon establishing the delegate.
- (void) - deviceDisconnected

  Fires when device disconnects.
- (void) - plugStatusChange:
- (void) - dataInOutMonitor:incoming:
- (void) - swipeMSRData:
- (void) - deviceMessage:
- (void) - eventFunctionICC:

11.3.1 Detailed Description

Protocol methods established for IDT_UniPay class

11.3.2 Method Documentation

11.3.2.1 - (void) dataInOutMonitor: (NSData *) *data* incoming:(BOOL) *isIncoming* [optional]

All incoming/outgoing data going to the device can be monitored through this delegate.

Parameters

data	The serial data represented as a NSData object
isIncoming	The direction of the data • `TRUE` specifies data being received from the device, • `FALSE` indicates data being sent to the device.

11.3.2.2 - (void) deviceMessage: (NSString *) *message* [optional]

Receives messages from the framework

Parameters

message	String message transmitted by framework

11.3.2.3 - (void) eventFunctionICC: (Byte) *nICC_Attached* [optional]

UniPay ICC Event This function will be called when an ICC is attached or detached from reader. Applies to UniPay only

Parameters

nICC_Attached	Can be one of the following values: • 0x01: ICC attached while reader is idle • 0x00: ICC detached while reader is idle • 0x11: ICC attached while reader is in MSR mode • 0x10: After ICC Powered On, ICC Card Removal,Power off ICC

```
-(void) eventFunctionICC: (Byte) nICC_Attached
{
switch (nICC_Attached) {
case 0x01:
case 0x11:
{
LOGI(@"ICC event: ICC attached.");
}
break;

case 0x00:
case 0x10:
{
LOGI(@"ICC event: ICC detached.");
}
break;
}
}
```

11.3.2.4 - (void) plugStatusChange: (BOOL) *deviceInserted* [optional]

Monitors the headphone jack for device insertion/removal.

Parameters

deviceInserted	TRUE = device inserted, FALSE = device removed

11.3.2.5 - (void) swipeMSRData: (IDTMSRData ∗) *cardData* [optional]

Receives card data from MSR swipe.

Parameters

cardData	Captured card data from MSR swipe

The documentation for this protocol was generated from the following file:

- SourceMac/IDT_UniPay.h

11.4 PowerOnStructure Struct Reference

`#include <IDTCommon.h>`

Data Fields

- BOOL sendIFS

  Send S(IFS) request if T=1 protocolError: Reference source not found.
- BOOL explicitPPS

  Explicit PPSError: Reference source not found.
- BOOL disableAutoPPS

  No auto pps for negotiate mode.
- BOOL disableResponseCheck

  No check on response of S(IFS) request.
- unsigned char * pps

  pps is used to set the Protocol and Parameters Selection between card and reader, only Di <= 4 are supported. pps must follow the structure specified in ISO 7816-3 as PPS0, [PPS1], [PPS2], and [PPS3]. For more information see ISO 7816-3 section 7.2.
- unsigned char ppsLength

  lenght of pps data

11.4.1 Detailed Description

Structure to set ICC power on options. Used by IDT_BTPay::icc_powerOnICC:response:() IDT_UniPay::icc_↩ powerOnICC:response:()

The documentation for this struct was generated from the following file:

- SourceMac/IDTCommon.h

Index