



# **iOS/OSX SDK Guide for UniPay 1.5**

**#80131532-001**

**Rev. A**



## Revision History

Revision	Description and Reason for Change	Date
A	Initial Release - Manual;User;UniPay1.5;SDK;iOS	1/11/2016
A	Updated Swift Implementation Information	7/12/2016

# Contents

<b>1</b>	<b>IDTech iOS SDK Reference Guide for UniPay 1.5</b>	<b>1</b>
<b>2</b>	<b>Important Security Notice</b>	<b>3</b>
2.1	Applicability . . . . .	3
2.2	What Does PA-DSS Mean to You? . . . . .	3
2.3	Third Party Applications . . . . .	4
2.4	PA-DSS Guidelines . . . . .	4
2.5	More Information . . . . .	9
<b>3</b>	<b>UniPay 1.5 Main Transaction Commands</b>	<b>11</b>
3.1	EMV Methods . . . . .	11
3.2	MSR . . . . .	12
<b>4</b>	<b>EMV Callback</b>	<b>13</b>
<b>5</b>	<b>Sending Direct Commands</b>	<b>14</b>
<b>6</b>	<b>Core Implementation UniPay 1.5: Objective-C</b>	<b>15</b>
6.1	Integrating with IDTech framework . . . . .	15
6.2	Import the necessary framework/libraries . . . . .	15
6.3	Add Import statements to utilize frameworks . . . . .	21
6.4	Amend the view controller interface to include the framework delegate classes: . . . . .	21
6.5	Implement any/all of the optional delegate protocols used to receive data from IDT_UniPayI_V Delegate: . . . . .	21
6.6	Call the Singleton instance of the IDT_UniPayI_V framework object: . . . . .	22
6.7	Sample Project Tutorial . . . . .	22
6.7.1	Step 1: Create New Project . . . . .	22
6.7.2	Step 2: Import Frameworks . . . . .	23
6.7.3	Step 3: Design Interface . . . . .	23
6.7.4	Step 4: Configure Header File . . . . .	24
6.7.5	Step 5: Configure Method File . . . . .	28
<b>7</b>	<b>Core Implementation UniPay 1.5: Swift</b>	<b>33</b>
7.1	Integrating with IDTech Framework . . . . .	33

7.2	Import the Necessary Framework/Libraries . . . . .	33
7.3	Create a Bridging Header File . . . . .	39
7.4	Add Import Statement to the Bridging Header File . . . . .	44
7.5	Amend the View Controller Interface . . . . .	44
7.6	Implement Optional Delegate Protocols . . . . .	44
7.7	Allocate/Initialize IDT_UniPayI_V Object . . . . .	45
7.8	Sample Project Tutorial . . . . .	45
7.8.1	Step 1: Create New Project . . . . .	46
7.8.2	Step 2: Import Frameworks . . . . .	47
7.8.3	Step 3: Design Interface . . . . .	47
7.8.4	Step 4: Configure the Bridging Header and View Controller Files . . . . .	48
7.8.5	Step 5: Finalize the View Controller File . . . . .	52
<b>8</b>	<b>UniPay 1.5 Error Code Reference</b>	<b>57</b>
<b>9</b>	<b>Enumeration Reference</b>	<b>59</b>
<b>10</b>	<b>EMV Tag Reference</b>	<b>63</b>
<b>11</b>	<b>LCD Foreign Language Mapping Table</b>	<b>72</b>
<b>12</b>	<b>Hierarchical Index</b>	<b>74</b>
12.1	Class Hierarchy . . . . .	74
<b>13</b>	<b>Class Index</b>	<b>75</b>
13.1	Class List . . . . .	75
<b>14</b>	<b>Class Documentation</b>	<b>76</b>
14.1	AIDEntry Struct Reference . . . . .	76
14.1.1	Detailed Description . . . . .	76
14.2	APDUResponse Class Reference . . . . .	76
14.2.1	Detailed Description . . . . .	77
14.2.2	Method Documentation . . . . .	77
14.2.2.1	clear() . . . . .	77
14.2.2.2	sharedController() . . . . .	77
14.3	ApplicationID Struct Reference . . . . .	77
14.3.1	Detailed Description . . . . .	79
14.4	CAKey Struct Reference . . . . .	79
14.4.1	Detailed Description . . . . .	79
14.5	CRLEntry Struct Reference . . . . .	79
14.5.1	Detailed Description . . . . .	80
14.6	ICCReaderStatus Struct Reference . . . . .	80
14.6.1	Detailed Description . . . . .	80

14.7 IDT_UniPayI_V Class Reference . . . . .	81
14.7.1 Detailed Description . . . . .	82
14.7.2 Method Documentation . . . . .	82
14.7.2.1 close() . . . . .	82
14.7.2.2 config_getSerialNumber:(NSString **response) . . . . .	82
14.7.2.3 device_cancelConnectToAudioReader() . . . . .	82
14.7.2.4 device_connectToAudioReader() . . . . .	82
14.7.2.5 device_connectToUSB() . . . . .	83
14.7.2.6 device_getFirmwareVersion:(NSString **response) . . . . .	83
14.7.2.7 device_getResponseCodeString:(int errorCode) . . . . .	83
14.7.2.8 device_isAudioReaderConnected() . . . . .	83
14.7.2.9 device_isConnected:(IDT_DEVICE_Types device) . . . . .	84
14.7.2.10 device_sendIDGCommand:subCommand:data:response:(unsigned char command,[subCommand] unsigned char subCommand,[data] NSData *data,[response] NSData **response) . . . . .	84
14.7.2.11 device_setAudioVolume:(float val) . . . . .	84
14.7.2.12 device_setPassThrough:(BOOL enablePassThrough) . . . . .	85
14.7.2.13 device_startRKI() . . . . .	85
14.7.2.14 emv_authenticateTransaction:(NSData *tags) . . . . .	86
14.7.2.15 emv_callbackResponseLCD:selection:(int mode,[selection] unsigned char selection) . . . . .	86
14.7.2.16 emv_completeOnlineEMVTransaction:hostResponseTags:(BOOL isSuccess,[hostResponseTags] NSData *tags) . . . . .	87
14.7.2.17 emv_disableAutoAuthenticateTransaction:(BOOL disable) . . . . .	88
14.7.2.18 emv_getEMVL2Version:(NSString **response) . . . . .	88
14.7.2.19 emv_removeApplicationData:(NSString *AID) . . . . .	88
14.7.2.20 emv_removeCAPK:index:(NSString *rid,[index] NSString *index) . . . . .	89
14.7.2.21 emv_removeCRLList() . . . . .	89
14.7.2.22 emv_removeTerminalData() . . . . .	90
14.7.2.23 emv_retrieveAIDList:(NSArray **response) . . . . .	90
14.7.2.24 emv_retrieveApplicationData:response:(NSString *AID,[response] NSDictionary **responseAID) . . . . .	90
14.7.2.25 emv_retrieveCAPK:index:response:(NSString *rid,[index] NSString *index,[response] CAKey **response) . . . . .	91
14.7.2.26 emv_retrieveCAPKFile:index:response:(NSString *rid,[index] NSString *index,[response] NSData **response) . . . . .	92
14.7.2.27 emv_retrieveCAPKList:(NSArray **response) . . . . .	92
14.7.2.28 emv_retrieveCRLList:(NSMutableArray **response) . . . . .	94
14.7.2.29 emv_retrieveTerminalData:(NSDictionary **responseData) . . . . .	94
14.7.2.30 emv_retrieveTransactionResult:retrievedTags:(NSData *tags,[retrievedTags] NSDictionary **retrievedTags) . . . . .	95
14.7.2.31 emv_setApplicationData:configData:(NSString *aidName,[configData] NSDictionary *data) . . . . .	95

14.7.2.32	emv_setCAPK:(CAKey key)	96
14.7.2.33	emv_setCAPKFile:(NSData *file)	97
14.7.2.34	emv_setCRLEntries:(NSData *data)	97
14.7.2.35	emv_setTerminalData:(NSDictionary *data)	98
14.7.2.36	emv_startTransaction:amtOther:type:timeout:tags:forceOnline:fallback:(double amount,[amtOther] double amtOther,[type] int type,[timeout] int timeout,[tags] NSData *tags,[forceOnline] BOOL forceOnline,[fallback] BOOL fallback)	98
14.7.2.37	icc_exchangeAPDU:response:(NSData *dataAPDU,[response] APDUResponse **response)	99
14.7.2.38	icc_getICCReaderStatus:(ICCReaderStatus **readerStatus)	99
14.7.2.39	icc_powerOffICC:(NSString **error)	100
14.7.2.40	icc_powerOnICC:(NSData **response)	100
14.7.2.41	isConnected()	100
14.7.2.42	msr_cancelMSRSwipe()	100
14.7.2.43	msr_startMSRSwipe()	101
14.7.2.44	SDK_version()	101
14.7.2.45	sharedController()	101
14.7.3	Property Documentation	101
14.7.3.1	delegate	101
14.8	<IDT_UniPayI_V_Delegate> Protocol Reference	101
14.8.1	Detailed Description	102
14.8.2	Method Documentation	102
14.8.2.1	dataInOutMonitor:incoming:(NSData *data,[incoming] BOOL isIncoming)	102
14.8.2.2	deviceMessage:(NSString *message)	102
14.8.2.3	emvTransactionData:errorCode:(IDTEMVData *emvData,[errorCode] int error)	102
14.8.2.4	lcdDisplay:lines:(int mode,[lines] NSArray *lines)	103
14.8.2.5	plugStatusChange:(BOOL deviceInserted)	103
14.8.2.6	swipeMSRData:(IDTMSRData *cardData)	103
14.9	IDTEMVData Class Reference	104
14.9.1	Detailed Description	104
14.9.2	Method Documentation	104
14.9.2.1	clear()	104
14.9.2.2	sharedController()	105
14.9.3	Property Documentation	105
14.9.3.1	resultCode	105
14.9.3.2	resultCodeV2	105
14.10	IDTMSRData Class Reference	106
14.10.1	Detailed Description	107
14.10.2	Method Documentation	107
14.10.2.1	clear()	107
14.10.2.2	sharedController()	108

14.10.3 Property Documentation . . . . .	108
14.10.3.1 captureEncodeType . . . . .	108
14.10.3.2 captureEncryptType . . . . .	108
14.10.3.3 event . . . . .	108
14.10.3.4 readStatus . . . . .	108
14.11 MaskAndEncryption Struct Reference . . . . .	109
14.11.1 Detailed Description . . . . .	109
14.11.2 Member Data Documentation . . . . .	109
14.11.2.1 encryptionOption . . . . .	109
14.11.2.2 maskOption . . . . .	109
14.12 PowerOnStructure Struct Reference . . . . .	110
14.12.1 Detailed Description . . . . .	110
14.13 TerminalData Struct Reference . . . . .	110
14.13.1 Detailed Description . . . . .	111
<b>Index</b>	<b>113</b>





## Chapter 1

# IDTech iOS SDK Reference Guide for UniPay 1.5



IDTech.framework is an Apple Framework that will be provided by IDTech as the main interface between iOS applications, the UniPay 1.5 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.

- [Core Implementation UniPay 1.5: Objective-C](#)
- [Core Implementation UniPay 1.5: Swift](#)
- [Important Security Notice](#)
- [UniPay 1.5 Main Transaction Commands](#)

- [EMV Callback](#)
- [Sending Direct Commands](#)
- [EMV Tag Reference](#)
- [Enumeration Reference](#)
- [UniPay 1.5 Error Code Reference](#)
- [LCD Foreign Language Mapping Table](#)

## 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 <sup>1</sup>	Yes	Yes <sup>1</sup>	No
	Service Code <sup>1</sup>	Yes	Yes <sup>1</sup>	No
	Expiration Date <sup>1</sup>	Yes	Yes <sup>1</sup>	No
Sensitive Authentication Data <sup>2</sup>	Full Magnetic Stripe Data <sup>3</sup>	No	N/A	N/A
	CAV2/CID/CVC2/CVV2	No	N/A	N/A
	PIN/PIN Block	No	N/A	N/A

<sup>1</sup> 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.

<sup>2</sup> Do not store sensitive authentication data after authorization (even if encrypted).

<sup>3</sup> 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 PA-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 data for 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, IEEE 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 SSID 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](mailto:capninfocenter@aexp.com)

## Chapter 3

# UniPay 1.5 Main Transaction Commands

The methods below are provided as a reference to the main commands needed to execute a contact or contactless EMV transaction, or collect MSR information from a swipe.

### 3.1 EMV Methods

#### Start EMV Transaction

```
IDT_UniPayI_V::emv_startTransaction:amtOther:type:timeout:tags:forceOnline:returnTags:fallback:()
```

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

```
IDT_UniPayI_V::emv_authenticateTransaction()
```

By default, auto-authenticate is ON and this step does not need to be performed. If auto-authenticate is OFF ([emv\\_disableAutoAuthenticateTransaction: \(IDT\\_UniPayI\\_V\)](#)), when the results come back as EMV\_RESULT\_C↔ODE.EMV\_RESULT\_CODE\_AUTHENTICATE\_TRANSACTION, this method must be called to continue the EMV transaction.

#### Complete Online EMV Transaction

```
IDT_UniPayI_V::emv_completeOnlineEMVTransaction:hostResponseTags:returnTags:()
```

After receiving a host response, pass host tags (minimum 8A Authorization Response Code) as a TLV stream through the tags parameter. EMV tags can be parsed returned pointer.

If there was a communication error with host, you must still finish the EMV transaction by passing "FALSE" for isSuccess, and nil for tags.

#### Terminal Configuration

```
emv_retrieveTerminalData: (IDT_UniPayI_V)
emv_removeTerminalData (IDT_UniPayI_V)
emv_setTerminalData: (IDT_UniPayI_V)
```

Methods for terminal configuration. When setting the terminal data, you populate and pass and TerminalFile structure.

#### AID Management

```
emv_retrieveApplicationData:response: (IDT_UniPayI_V)
emv_removeApplicationData: (IDT_UniPayI_V)
emv_setApplicationData:configData: (IDT_UniPayI_V)
```

[emv\\_retrieveAIDList: \(IDT\\_UniPayI\\_V\)](#)

Methods for AID management on Contact EMV. When setting the AID, you pass tags in TLV format. When retrieving AID, you can receive the results as tags in TLV format. When retrieving the AID list, the list of AID Names/length can be retrieved from the NSArray of NSString.

#### CAPK Management

[emv\\_retrieveCAPK:index:response: \(IDT\\_UniPayI\\_V\)](#)  
[emv\\_removeCAPK:index: \(IDT\\_UniPayI\\_V\)](#)  
[emv\\_setCAPK: \(IDT\\_UniPayI\\_V\)](#)  
[emv\\_retrieveCAPKList: \(IDT\\_UniPayI\\_V\)](#)

Methods for Certificate Authority Public Key management. When setting the CAPK, you populate and pass the key as a sequence of ordered bytes. When specifying a CAPK to retrieve or remove, you populate the name in the NSData parameter. When retrieving the CAPK list, the list of RID/Index can be retrieved from the ordered NSData stream, 6 bytes each, bytes 1-5 RID, byte 6 index.

#### CRL Management

[emv\\_retrieveCRLList: \(IDT\\_UniPayI\\_V\)](#)  
[emv\\_removeCRLList \(IDT\\_UniPayI\\_V\)](#)  
[emv\\_setCRLEntries: \(IDT\\_UniPayI\\_V\)](#)

Methods for Certificate Revocation List management.

#### APDU Communication

[device\\_setPassThrough: \(IDT\\_UniPayI\\_V\)](#)  
[icc\\_powerOnICC: \(IDT\\_UniPayI\\_V\)](#)  
[icc\\_powerOffICC: \(IDT\\_UniPayI\\_V\)](#)

[icc\\_exchangeAPDU:response: \(IDT\\_UniPayI\\_V\)](#)

Allows the direct sending of APDU packets to ICC. Pass through mode must first be enabled. Then Power On needs to complete successfully. Then APDU packet exchange can take place

## 3.2 MSR

#### Request Swipe

[msr\\_startMSRSwipe \(IDT\\_UniPayI\\_V\)](#)

#### Cancel Swipe

[msr\\_cancelMSRSwipe \(IDT\\_UniPayI\\_V\)](#)

## Chapter 4

# EMV Callback

During an EMV transaction, without a built-in LCD display on the UniPay 1.5, LCD Display messages will be returned as an EMV Callback.

In the available protocols for the [IDT\\_UniPayI\\_V](#) class, there is a protocol as follows:

```
- (void) lcdDisplay:(int)mode lines:(NSArray*)lines;
```

Once this delegate is implemented, if an EMV transaction requires information to be displayed on what would normally be an LCD display controlled by the Kernel, this data is returned with the display message type, and message string(s).

To evaluate what kind of LCD message, you interpret the mode as follows:

- 1- LCD\_DISPLAY\_MODE\_MENU: Menu selection, response required with selected menu index #, or 0 to cancel
- 2- LCD\_DISPLAY\_MODE\_PROMPT: Message Prompt, response required 'E' for Enter/Accept, or 'C' for cancel
- 3- LCD\_DISPLAY\_MODE\_MESSAGE: Display Message, no response required
- 8 - LCD\_DISPLAY\_MODE\_LANGUAGE\_SELECT: Language selection, response required with selected language index #
- 16 - LCD\_DISPLAY\_MODE\_CLEAR\_SCREEN: Request to clear LCD screen of information

If the mode is LCD\_DISPLAY\_MODE\_MESSAGE or LCD\_DISPLAY\_MODE\_CLEAR\_SCREEN, these do not pause the EMV transaction. These two modes are for displaying a message (no response required), or for clearing the screen.

If the mode is LCD\_DISPLAY\_MODE\_MENU, LCD\_DISPLAY\_MODE\_PROMPT, or LCD\_DISPLAY\_MODE\_LANGUAGE\_SELECT, the provided message must be displayed, and then the EMV transaction pauses until a response is sent to [emv\\_callbackResponseLCD:selection:](#) ([IDT\\_UniPayI\\_V](#)).

The message to display is returned as an NSArray of NSStrings. This contains either Message String, or a Message retrieved from the LCD Foreign Language Mapping Table ([Foreign Language Mapping Table](#)).

## Chapter 5

# Sending Direct Commands

The main purpose of IDTech.framework for UniPay 1.5 is to expedite integration to the device by providing the connectivity and communication protocols. It also provides the main functions to get device info, perform contact EMV Transactions, perform swipe transactions, and to modify contact EMV data files.

The UniPay 1.5 has an extensive and powerful command set based on the NEO platform. A NEO command consists of a Command, a Sub Command, and optionally data. To access these commands, please reference the NEO/IDG Command document included as a separate item in the SDK. Please note that Protocol 1 commands have been deprecated, and any existing Protocol 1 commands relevant to UniPay 1.5 can be accomplished by a Protocol 2 command. The IDTech.framework uses the following the command to send Protocol 2 commands to UniPay 1.5:

[device\\_sendIDGCommand:subCommand:data:response: \(IDT\\_UniPayI\\_V\)](#)

Any function not supported by the SDK can be sent with the sendIDGCommand.

## Chapter 6

# Core Implementation UniPay 1.5: Objective-C

IDTech Framework includes class libraries to interface with the UniPay 1.5. This guide assume a fair understanding of Xcode 5.0+ and general Apple iOS programming knowledge.

### 6.1 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 IDT\\_UniPayI\\_V objects](#)
- [Sample Project Tutorial](#)

### 6.2 Import the necessary framework/libraries

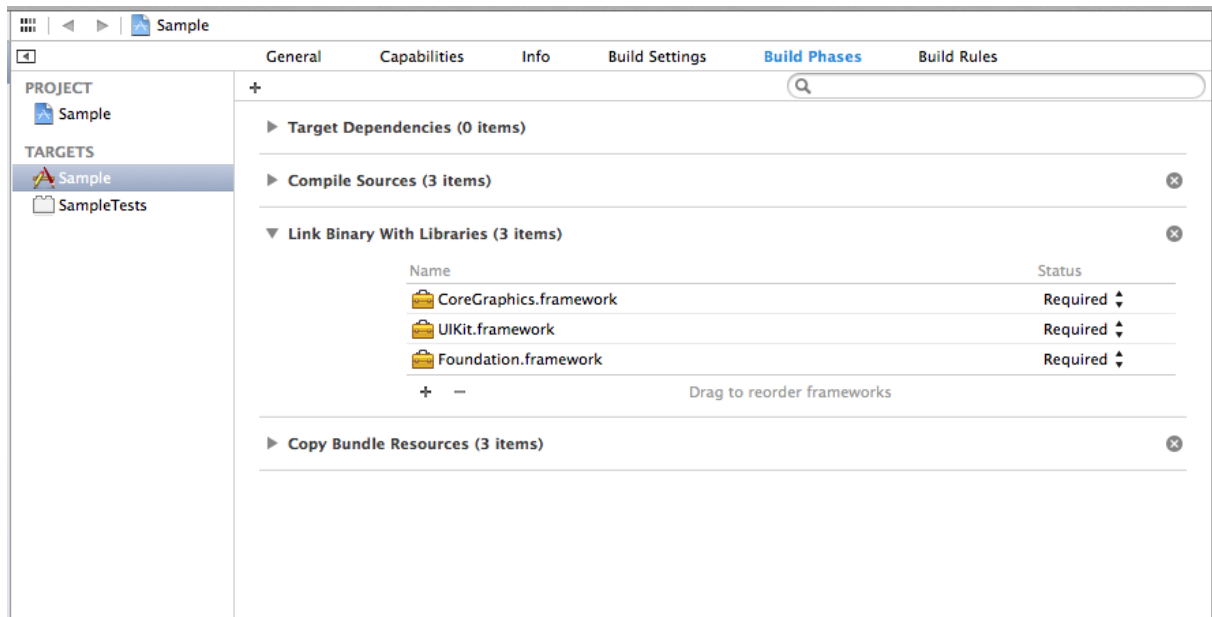
Communicating with IDTech Devices requires the following framework/libraries to be imported into the project:

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

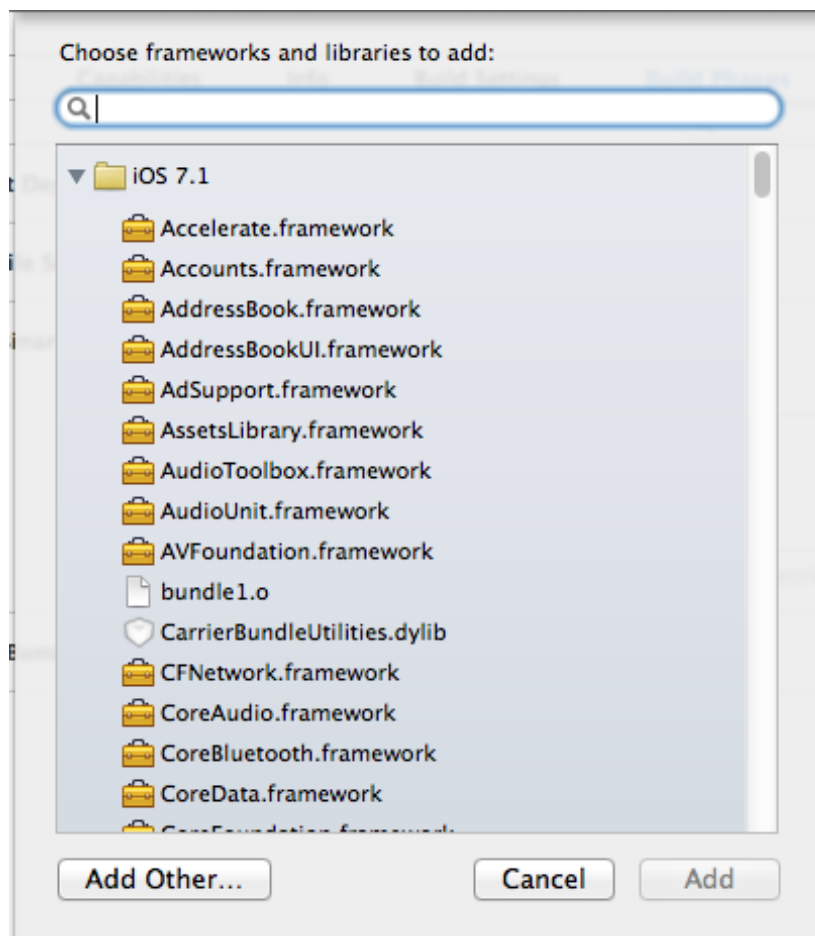
Also, import the following resource bundle:

- IDTech.bundle

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

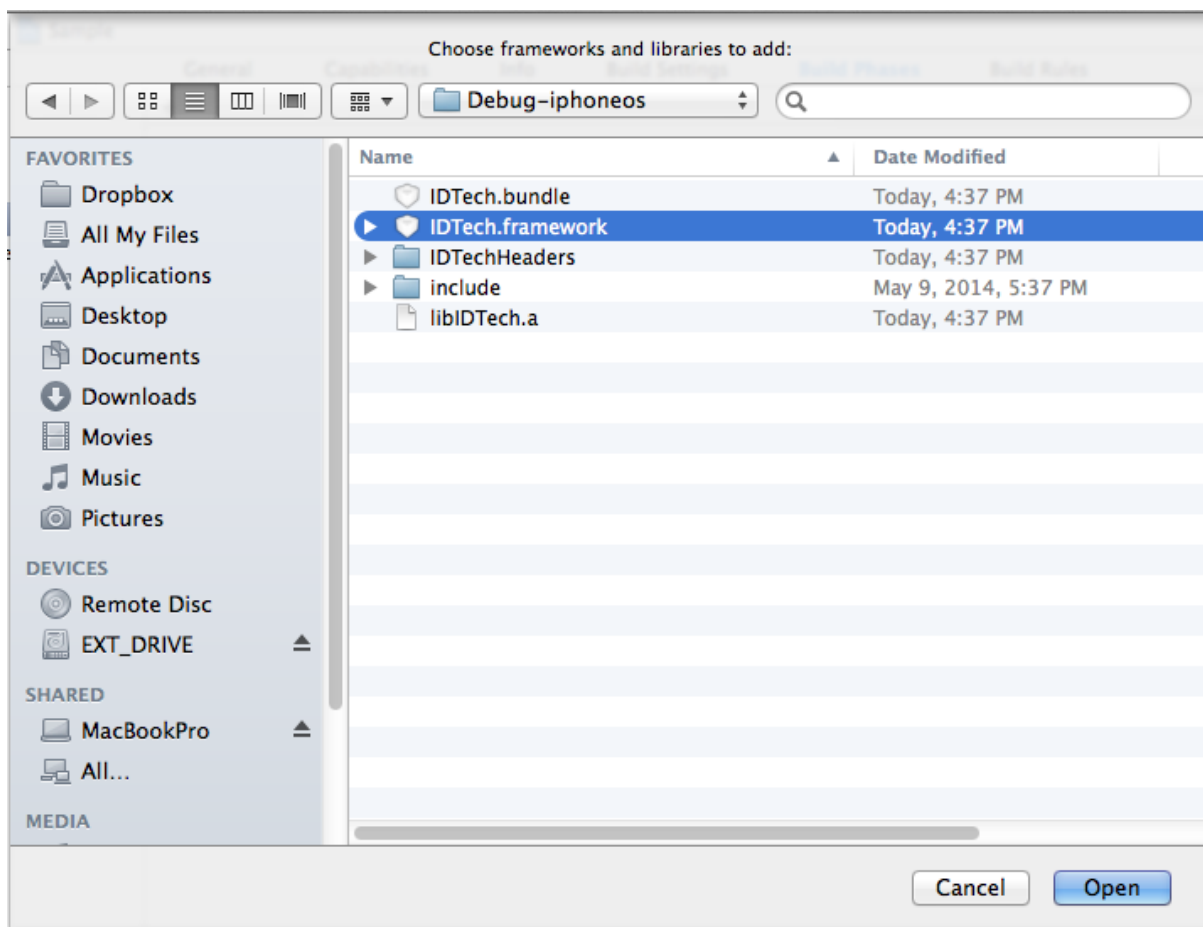


On the Choose Frameworks screen, click "Add Other" in the lower left

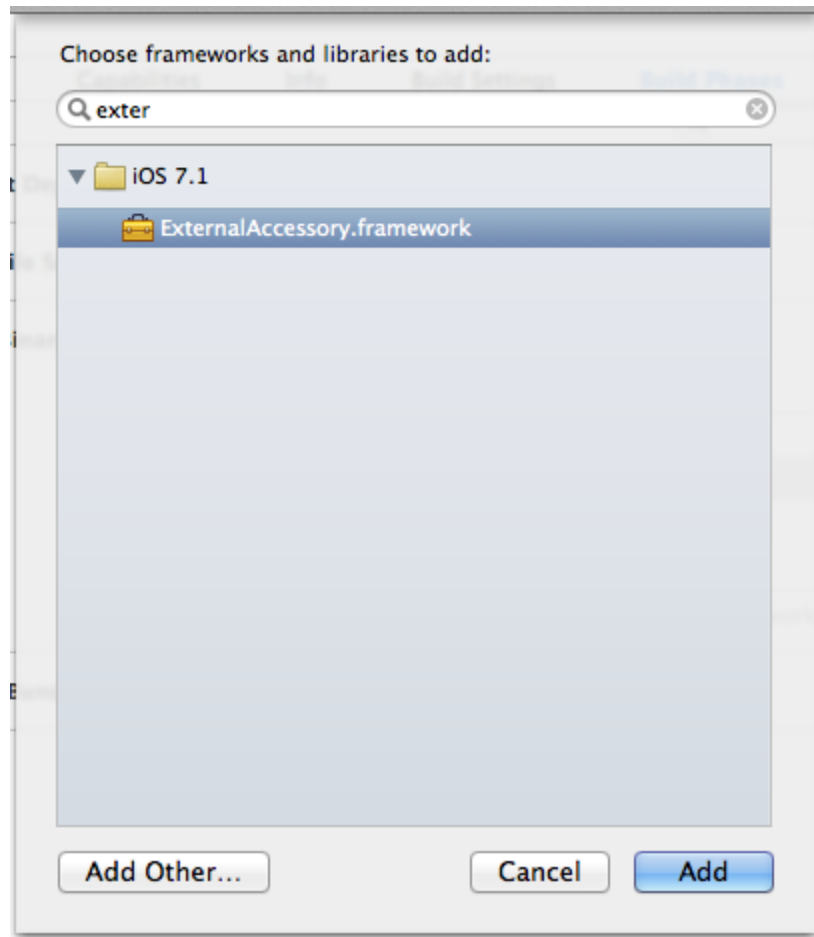


Navigate to the IDTech.framework folder, and click "Open"



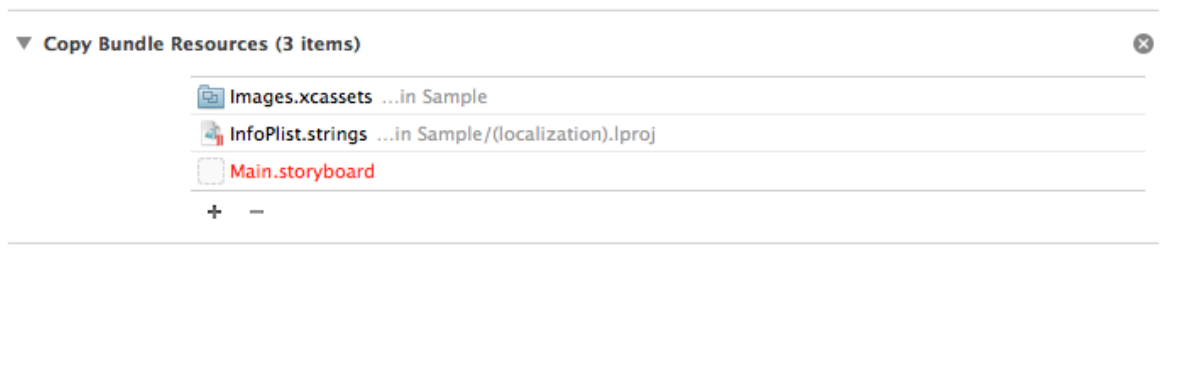


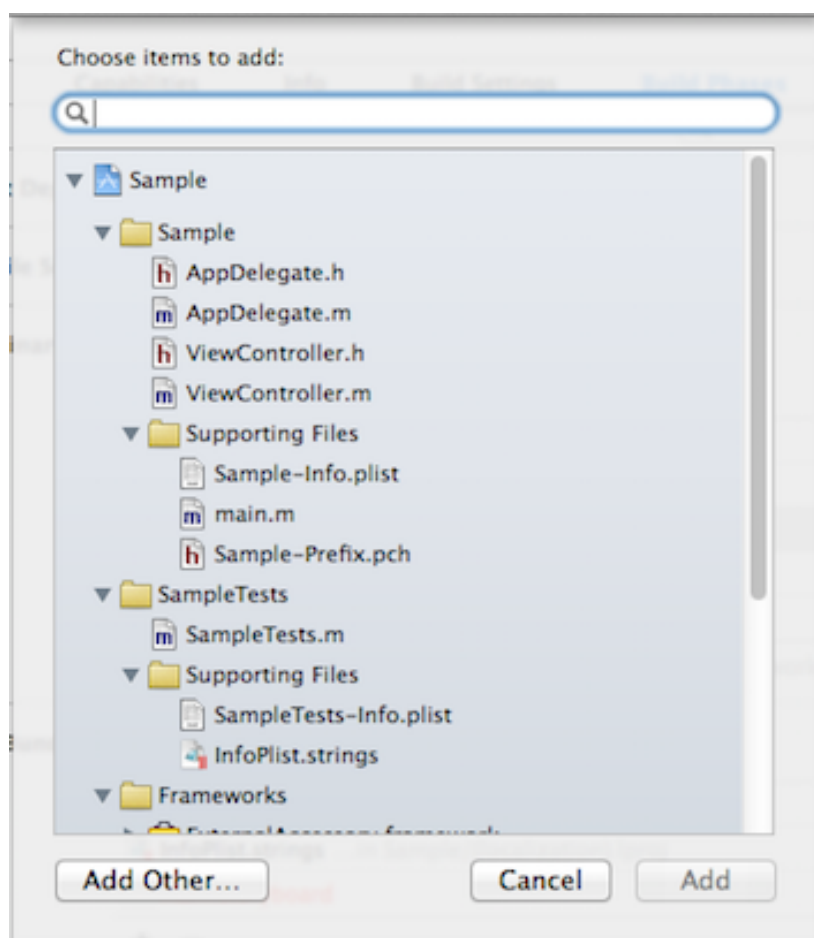
Link the ExternalAccessory framework. On the Choose Frameworks screen, type "exter" into the search bar, select ExternalAccessory.framework and click "Open"

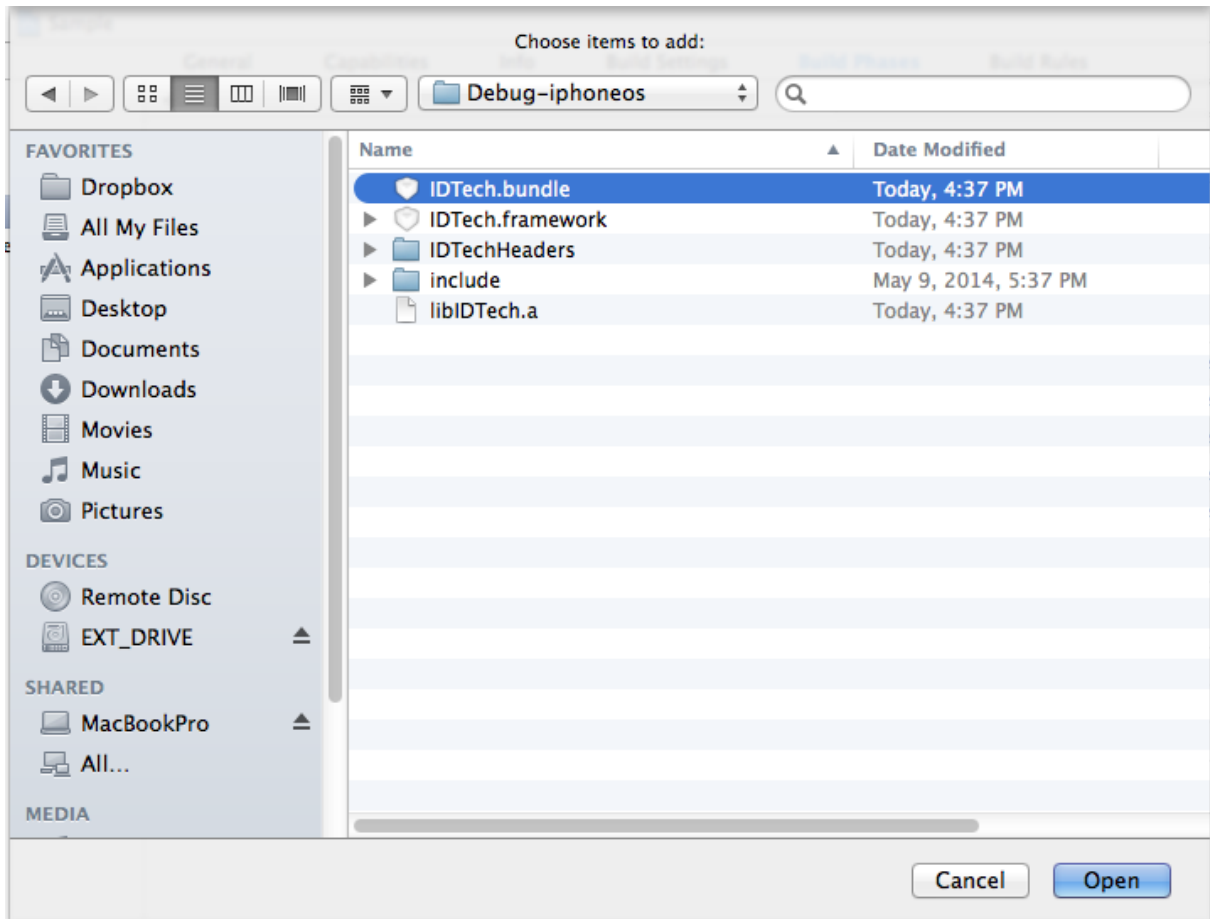


Repeat process for MediaPlayer.framework, AVFoundation.framework, AudioToolbox.framework, and CFNetwork.framework.

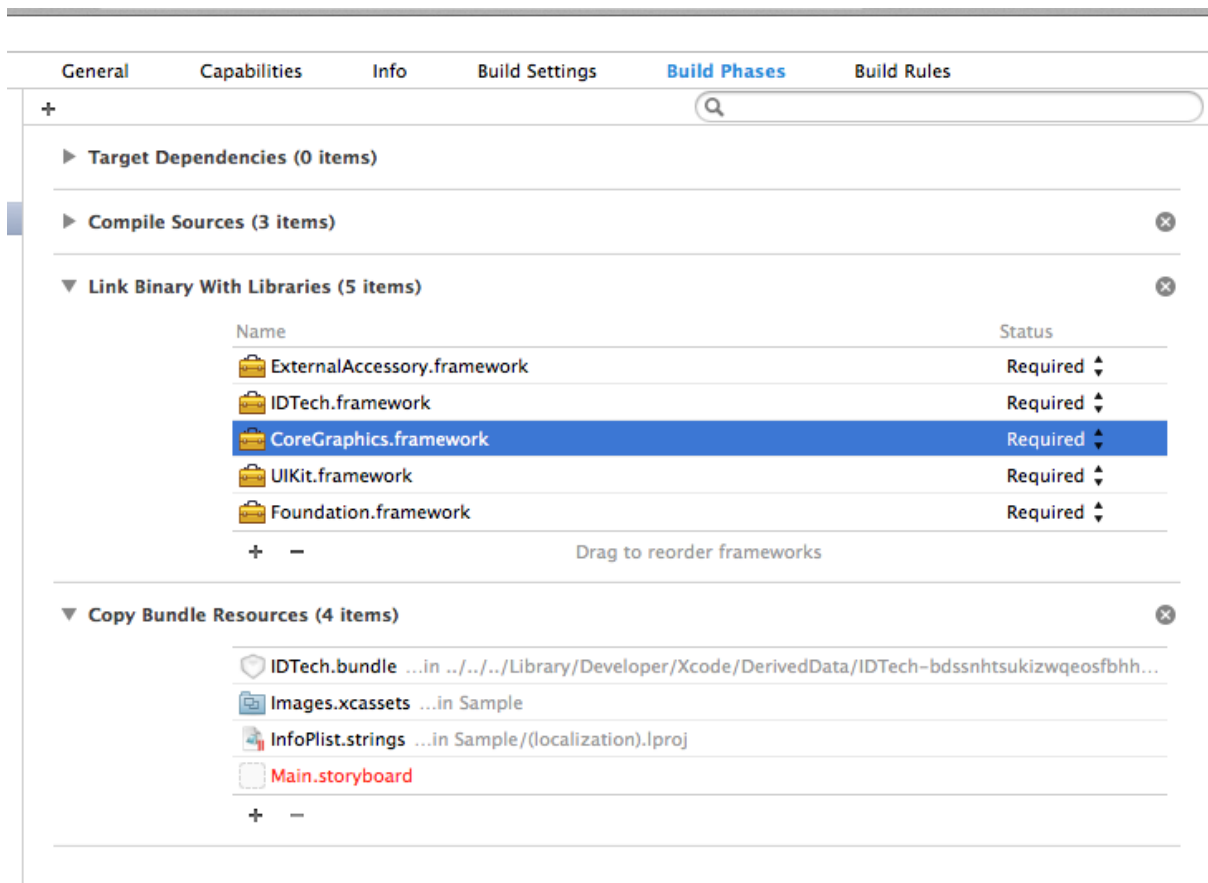
Link another library. Under Copy Bundle, click the Add (+) button, click "Add Other", navigate to and select the IDTech.bundle file and click "Open"







The Build Phases should now include the required frameworks/libraries for the UniPay 1.5



## 6.3 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>
```

## 6.4 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_UniPayI_V_Delegate>
```

## 6.5 Implement any/all of the optional delegate protocols used to receive data from IDT\_UniPayI\_V\_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) lcdDisplay:(int) mode lines:(NSArray*) lines;

- (void) emvTransactionData:(IDTEMVData*) emvData errorCode:(int) error;

```

## 6.6 Call the Singleton instance of the IDT\_UniPayI\_V framework object:

A Singleton instance has been established in the [IDT\\_UniPayI\\_V](#) 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_UniPayI_V sharedInstance] setDelegate:self];
}

```

## 6.7 Sample Project Tutorial

Using Xcode 6.4, we will create a sample project that will interface with the UniPay 1.5 and will perform the following activities:

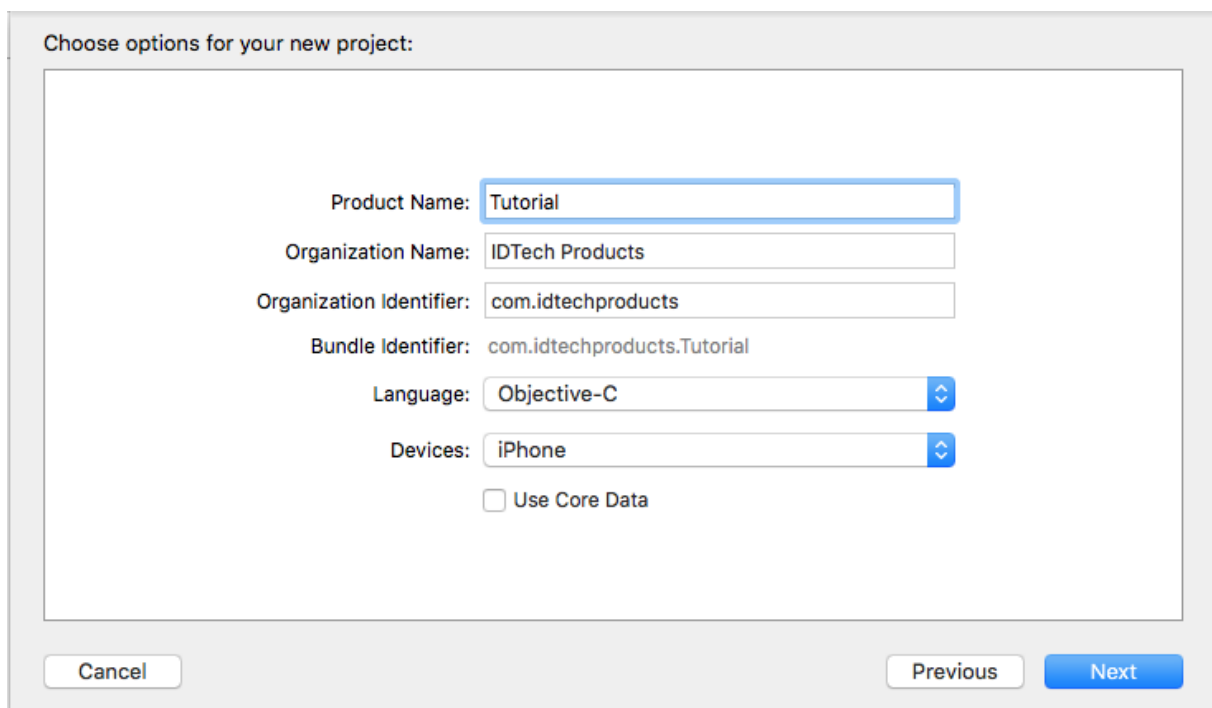
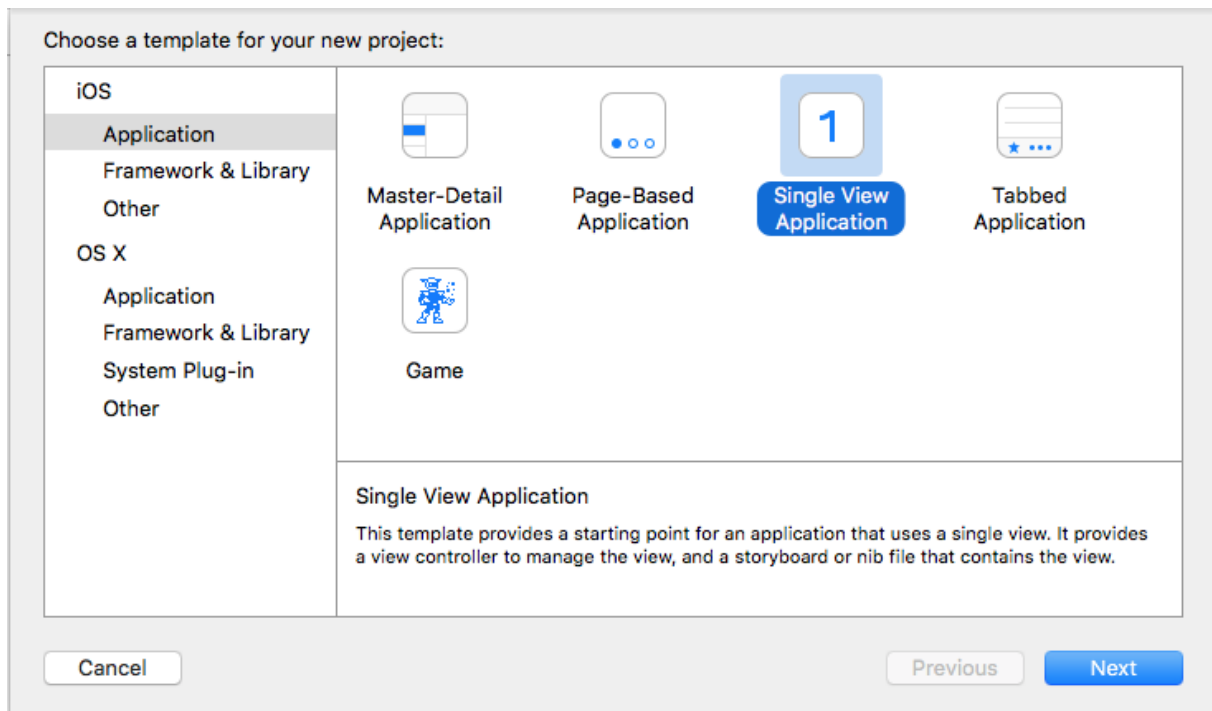
- Auto-Connect and display connection status
- Get Device Firmware
- Start/Stop Transaction Request for MSR
- Start/Complete/Cancel EMV Transaction
- Show LCD Display for EMV transaction
- Automatically select first AID or first Language if prompted

Protocol Delegates:

- Delegate to receive card swipes
- Delegate to detect headphone plug changes
- Delegate to detect device connected
- Delegate to detect device disconnected
- Delegate to receive EMV tag data

### 6.7.1 Step 1: Create New Project

Create a new Single View Application in Xcode



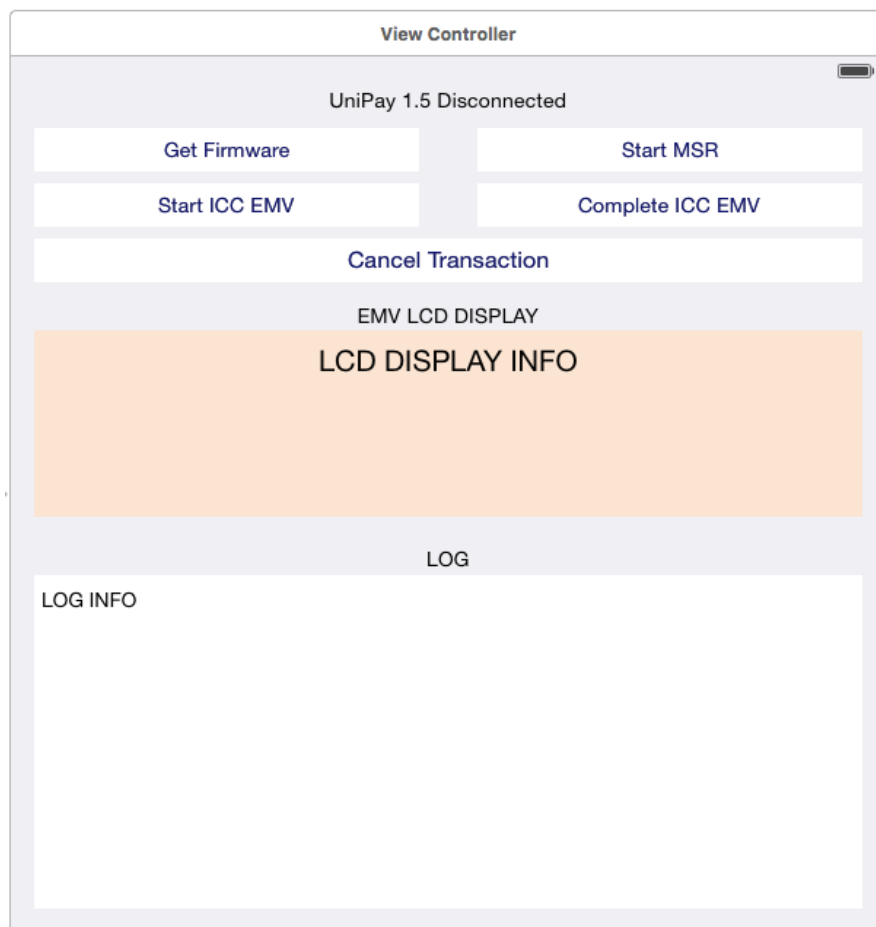
### 6.7.2 Step 2: Import Frameworks

[Import the necessary framework/libraries](#)

### 6.7.3 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 text views to communicate data from the UniPay 1.5 and for EMV LCD display information. 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 MSR
  - Start ICC EMV
  - Complete ICC EMV
  - Cancel Transaction (add constraints accordingly so layout maps to intended screen size)



#### 6.7.4 Step 4: Configure Header File

In the header file, perform the following:

- [Add Import statements to utilize frameworks](#)
- [Amend the view controller interface](#)
- Create an IBOutlet for the two 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 4 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_UniPayI_V_Delegate>
{
    IBOutlet UITextView *lcdTextView;
    IBOutlet UITextView *logTextView;
    IBOutlet UILabel *connectedLabel;
}

-(IBAction) getFirmware:(id)sender;
-(IBAction) startMSR_CTLS:(id)sender;
-(IBAction) startEMV:(id)sender;
-(IBAction) completeEMV:(id)sender;
-(IBAction) cancelTransaction:(id)sender;

@property(n nonatomic, strong) UITextView *lcdTextView;
@property(n nonatomic, strong) UITextView *logTextView;
@property(n nonatomic, strong) UILabel *connectedLabel;

@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="7706"
    systemVersion="15B42" targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES"
    useTraitCollections="YES" initialViewController="vXZ-lx-hvc">
    <dependencies>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="7703"/>
        <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" misplaced="YES" text="UniPay 1.5 Disconnected"
                                textAlignment="center" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
                                translatesAutoresizingMaskIntoConstraints="NO" id="eDI-cL-7k6">
                                    <rect key="frame" x="16" y="20" width="568" height="21"/>
                                    <animations/>
                                    <fontDescription key="fontDescription" type="system" pointSize="14"/>
                                    <color key="textColor" cocoaTouchSystemColor="darkTextColor"/>
                                    <nil key="highlightedColor"/>
                                </label>
                            <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
                                horizontalHuggingPriority="251" verticalHuggingPriority="251" ambiguous="YES" misplaced="YES" text="" lineBreakMode="
                                tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
                                translatesAutoresizingMaskIntoConstraints="NO" id="BNT-fB-6Ni">
                                    <rect key="frame" x="288" y="53" width="24" height="21"/>
                                    <animations/>
                                    <fontDescription key="fontDescription" type="system" pointSize="17"/>
                                    <color key="textColor" cocoaTouchSystemColor="darkTextColor"/>
                                    <nil key="highlightedColor"/>
                                </label>
                            <button opaque="NO" contentMode="scaleToFill" misplaced="YES"
                                contentHorizontalAlignment="center" contentVerticalAlignment="center" lineBreakMode="middleTruncation"
                                translatesAutoresizingMaskIntoConstraints="NO" id="8zA-5c-Igm">
                                    <rect key="frame" x="16" y="49" width="264" height="30"/>
                                    <animations/>
                                    <color key="backgroundColor" red="1" green="1" blue="1" alpha="1"
                                        colorSpace="calibratedRGB"/>
                                    <fontDescription key="fontDescription" type="system" pointSize="14"/>
                                    <state key="normal" title="Get Firmware">
                                        <color key="titleColor" red="0.068003949080000001" green="
                                        0.072250786509999998" blue="0.44422978940000002" alpha="1" colorSpace="calibratedRGB"/>
                                        <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
                                        calibratedWhite"/>
                                    </state>
                                </button>
                        </subviews>
                    </view>
                </viewController>
            </objects>
            <connections>
```

```

        <action selector="getFirmware:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="ONf-1N-HDd"/>
    </connections>
</button>
    <button opaque="NO" contentMode="scaleToFill" misplaced="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" lineBreakMode="middleTruncation"
translateAutoresizingMaskIntoConstraints="NO" id="QvK-Hw-R0Z">
    <rect key="frame" x="320" y="49" width="264" height="30"/>
    <animations/>
    <color key="backgroundColor" red="1" green="1" blue="1" alpha="1"
colorSpace="calibratedRGB"/>
    <fontDescription key="fontDescription" type="system" pointSize="14"/>
    <state key="normal" title="Start MSR">
        <color key="titleColor" red="0.068003949080000001" green="
0.072250786509999998" blue="0.44422978940000002" alpha="1" colorSpace="calibratedRGB"/>
        <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
    </state>
    <connections>
        <action selector="startMSR_CTLs:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="T65-oV-nQ4"/>
    </connections>
</button>
    <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
horizontalHuggingPriority="251" verticalHuggingPriority="251" ambiguous="YES" misplaced="YES" text="" lineBreakMode="
tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translateAutoresizingMaskIntoConstraints="NO" id="c2A-gb-32n">
    <rect key="frame" x="288" y="91" width="24" height="21"/>
    <animations/>
    <fontDescription key="fontDescription" type="system" pointSize="17"/>
    <color key="textColor" cocoaTouchSystemColor="darkTextColor"/>
    <nil key="highlightedColor"/>
</label>
    <button opaque="NO" contentMode="scaleToFill" misplaced="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" lineBreakMode="middleTruncation"
translateAutoresizingMaskIntoConstraints="NO" id="KML-Tf-QdE">
    <rect key="frame" x="16" y="87" width="264" height="30"/>
    <animations/>
    <color key="backgroundColor" red="1" green="1" blue="1" alpha="1"
colorSpace="calibratedRGB"/>
    <fontDescription key="fontDescription" type="system" pointSize="14"/>
    <state key="normal" title="Start ICC EMV">
        <color key="titleColor" red="0.068003949080000001" green="
0.072250786509999998" blue="0.44422978940000002" alpha="1" colorSpace="calibratedRGB"/>
        <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
    </state>
    <connections>
        <action selector="startEMV:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="jXs-Hw-Y20"/>
    </connections>
</button>
    <button opaque="NO" contentMode="scaleToFill" misplaced="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" lineBreakMode="middleTruncation"
translateAutoresizingMaskIntoConstraints="NO" id="U0Y-vt-9An">
    <rect key="frame" x="16" y="125" width="568" height="30"/>
    <animations/>
    <color key="backgroundColor" red="1" green="1" blue="1" alpha="1"
colorSpace="calibratedRGB"/>
    <fontDescription key="fontDescription" type="system" pointSize="16"/>
    <state key="normal" title="Cancel Transaction">
        <color key="titleColor" red="0.068003949080000001" green="
0.072250786509999998" blue="0.44422978940000002" alpha="1" colorSpace="calibratedRGB"/>
        <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
    </state>
    <connections>
        <action selector="cancelTransaction:" destination="vXZ-lx-hvc"
eventType="touchUpInside" id="rOr-8n-gJz"/>
    </connections>
</button>
    <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
scaleToFill" ambiguous="YES" misplaced="YES" editable="NO" text="LCD DISPLAY INFO" textAlignment="center"
translateAutoresizingMaskIntoConstraints="NO" id="XfN-t5-Cxv">
    <rect key="frame" x="16" y="188" width="568" height="128"/>
    <animations/>
    <color key="backgroundColor" red="0.98823529480000005" green="
0.89411765340000005" blue="0.8156862855" alpha="1" colorSpace="calibratedRGB"/>
    <fontDescription key="fontDescription" name="HelveticaNeue" family="
Helvetica Neue" pointSize="20"/>
    <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
</textView>
    <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
scaleToFill" ambiguous="YES" misplaced="YES" editable="NO" text="LOG INFO" translatesAutoresizingMaskIntoConstraints="
NO" id="Ybs-ph-PwR">
    <rect key="frame" x="16" y="356" width="568" height="229"/>
    <animations/>

```

```

        <color key="backgroundColor" white="1" alpha="1" colorSpace="
calibratedWhite"/>
        <fontDescription key="fontDescription" name="HelveticaNeue" family="
Helvetica Neue" pointSize="14"/>
        <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
    </textView>
    <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
horizontalHuggingPriority="251" verticalHuggingPriority="251" misplaced="YES" text="EMV LCD DISPLAY" textAlignment="
center" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="oIl-9l-J9y">
        <rect key="frame" x="16" y="168" width="568" height="21"/>
        <animations/>
        <fontDescription key="fontDescription" type="system" pointSize="14"/>
        <color key="textColor" cocoaTouchSystemColor="darkTextColor"/>
        <nil key="highlightedColor"/>
    </label>
    <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
horizontalHuggingPriority="251" verticalHuggingPriority="251" ambiguous="YES" misplaced="YES" text="LOG" textAlignment="
center" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="ngg-AN-FNd">
        <rect key="frame" x="16" y="335" width="568" height="21"/>
        <animations/>
        <fontDescription key="fontDescription" type="system" pointSize="14"/>
        <color key="textColor" cocoaTouchSystemColor="darkTextColor"/>
        <nil key="highlightedColor"/>
    </label>
    <button opaque="NO" contentMode="scaleToFill" misplaced="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="IhF-UM-clE">
        <rect key="frame" x="320" y="87" width="264" height="30"/>
        <animations/>
        <color key="backgroundColor" red="1" green="1" blue="1" alpha="1"
colorSpace="calibratedRGB"/>
        <fontDescription key="fontDescription" type="system" pointSize="14"/>
        <state key="normal" title="Complete ICC EMV">
            <color key="titleColor" red="0.068003949080000001" green="
0.072250786509999998" blue="0.44422978940000002" alpha="1" colorSpace="calibratedRGB"/>
            <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
        </state>
        <connections>
            <action selector="completeEMV:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="sBg-6w-7jS"/>
        </connections>
    </button>
</subviews>
<animations/>
<color key="backgroundColor" cocoaTouchSystemColor="groupTableViewBackgroundColor"/
>

    <constraints>
        <constraint firstItem="ngg-AN-FNd" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="07N-fx-7L7"/>
        <constraint firstItem="Ybs-ph-PwR" firstAttribute="top" secondItem="ngg-AN-FNd"
secondAttribute="bottom" id="0R5-b0-UV6"/>
        <constraint firstItem="IhF-UM-clE" firstAttribute="top" secondItem="QvK-Hw-R0Z"
secondAttribute="bottom" constant="8" id="3wT-WY-O6y"/>
        <constraint firstItem="KML-Tf-QdE" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="4xE-db-nPj"/>
        <constraint firstItem="XfN-t5-Cxv" firstAttribute="top" secondItem="U0Y-vt-9An"
secondAttribute="bottom" constant="33" id="7QY-kl-X8y"/>
        <constraint firstItem="eDI-cL-7k6" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="7fb-Cp-cCv"/>
        <constraint firstItem="QvK-Hw-R0Z" firstAttribute="top" secondItem="8zA-5c-Igm"
secondAttribute="bottom" constant="8" id="8tx-HW-5Nd"/>
        <constraint firstItem="oIl-9l-J9y" firstAttribute="top" secondItem="U0Y-vt-9An"
secondAttribute="bottom" constant="13" id="9kb-SL-MjC"/>
        <constraint firstItem="8zA-5c-Igm" firstAttribute="top" secondItem="eDI-cL-7k6"
secondAttribute="bottom" constant="8" id="9p8-gV-NRe"/>
        <constraint firstItem="QvK-Hw-R0Z" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="AJW-HY-0Gz"/>
        <constraint firstItem="oIl-9l-J9y" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="Azf-64-uHX"/>
        <constraint firstAttribute="centerX" secondItem="c2A-gb-32n" secondAttribute="
centerX" id="CIj-z7-200"/>
        <constraint firstItem="2fi-mo-0CV" firstAttribute="top" secondItem="Ybs-ph-PwR"
secondAttribute="bottom" constant="15" id="Hr6-bD-4h3"/>
        <constraint firstItem="ngg-AN-FNd" firstAttribute="top" secondItem="XfN-t5-Cxv"
secondAttribute="bottom" constant="19" id="Hvp-Bq-Jmg"/>
        <constraint firstItem="8zA-5c-Igm" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="KNU-4P-TIA"/>
        <constraint firstItem="eDI-cL-7k6" firstAttribute="top" secondItem="jyV-Pf-zRb"
secondAttribute="bottom" id="NTQ-83-15F"/>
        <constraint firstItem="IhF-UM-clE" firstAttribute="leading" secondItem="
c2A-gb-32n" secondAttribute="trailing" constant="8" id="P4V-QW-bw0"/>
        <constraint firstAttribute="centerX" secondItem="BNt-fB-6Ni" secondAttribute="
centerX" id="RYY-mX-lui"/>
        <constraint firstItem="Ybs-ph-PwR" firstAttribute="leading" secondItem="

```

```

kh9-bI-dsS" secondAttribute="leadingMargin" id="W3k-et-s0A"/>
    <constraint firstItem="c2A-gb-32n" firstAttribute="leading" secondItem="
KML-Tf-QdE" secondAttribute="trailing" constant="8" id="WHf-cP-6l6"/>
    <constraint firstItem="eDI-cL-7k6" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="WeO-Q4-18C"/>
    <constraint firstItem="U0Y-vt-9An" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="egE-ht-XL2"/>
    <constraint firstItem="XfN-t5-Cxv" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="eoS-jT-tfR"/>
    <constraint firstItem="Ybs-ph-PwR" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="gxc-v5-oE5"/>
    <constraint firstItem="QvK-Hw-R0Z" firstAttribute="leading" secondItem="
BNt-fB-6Ni" secondAttribute="trailing" constant="8" id="jaT-3B-urM"/>
    <constraint firstItem="U0Y-vt-9An" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="l4a-eF-pGd"/>
    <constraint firstItem="QvK-Hw-R0Z" firstAttribute="top" secondItem="eDI-cL-7k6"
secondAttribute="bottom" constant="8" id="pAi-z4-TRm"/>
    <constraint firstItem="XfN-t5-Cxv" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="pBl-HM-a2m"/>
    <constraint firstItem="ngg-AN-FNd" firstAttribute="leading" secondItem="
kh9-bI-dsS" secondAttribute="leadingMargin" id="qLF-YA-Iay"/>
    <constraint firstItem="oIl-9l-J9y" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="s28-T7-lrr"/>
    <constraint firstItem="U0Y-vt-9An" firstAttribute="top" secondItem="IhF-UM-clE"
secondAttribute="bottom" constant="8" id="wOn-ol-vv0"/>
    <constraint firstItem="IhF-UM-clE" firstAttribute="trailing" secondItem="
kh9-bI-dsS" secondAttribute="trailingMargin" id="wn6-dg-B6v"/>
    <constraint firstItem="BNt-fB-6Ni" firstAttribute="leading" secondItem="
8zA-5c-Igm" secondAttribute="trailing" constant="8" id="xXS-U6-Voz"/>
</constraints>
</view>
<connections>
    <outlet property="connectedLabel" destination="eDI-cL-7k6" id="EYe-RB-KZq"/>
    <outlet property="lcdTextView" destination="XfN-t5-Cxv" id="aN5-cG-h6Y"/>
    <outlet property="logTextView" destination="Ybs-ph-PwR" id="EOh-F5-kzy"/>
</connections>
</viewController>
<placeholder placeholderIdentifier="IBFirstResponder" id="x5A-6p-PRh" sceneMemberID="
firstResponder"/>
</objects>
</scene>
</scenes>
</document>

```

### 6.7.5 Step 5: Configure Method File

In the header file, perform the following:

- set delegate and initialize `IDT_UniPayI_V` singleton object in the `viewDidLoad` method. Reference: [Call the Singleton instance of the IDT\\_UniPayI\\_V framework object](#)

```

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

```

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

```

- (void) deviceConnected{
    [connectedLabel setText:@"UNIPAY 1.5 CONNECTED"];
}
- (void) deviceDisconnected{
    [connectedLabel setText:@"UNIPAY 1.5 DISCONNECTED"];
}

```

- Implement protocol delegate `swipeMSRData()` to receive card swipe data. Reference: [Implement optional delegate protocols](#)

```

- (void) appendMessageToResults:(NSString*) message{
    [logTextView setText:[NSString stringWithFormat:@"%s\n=====\n%",message, self.
logTextView.text]];
}

```

```

    }

    - (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:@"Encode Type: %@", @"ISO/ABA"]]
                            break;
                        case CAPTURE_ENCODE_TYPE_AAMVA:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encode Type: %@", @"AA/MVA"]]
                            break;
                        case CAPTURE_ENCODE_TYPE_Other:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encode Type: %@", @"Other"]]
                            break;
                        case CAPTURE_ENCODE_TYPE_Raw:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encode Type: %@", @"Raw"]]
                            break;
                        case CAPTURE_ENCODE_TYPE_JIS_I:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encode Type: %@", @"CAPTURE_ENCODE_TYPE_JIS_I"]]];
                            break;
                        case CAPTURE_ENCODE_TYPE_JIS_II:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encode Type: %@", @"CAPTURE_ENCODE_TYPE_JIS_II"]]];
                            break;
                        default:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encode Type: %@", @"UNKNOWN"]]
                            break;
                    }
                    switch (cardData.captureEncryptType) {
                        case CAPTURE_ENCRYPT_TYPE_AES:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encrypt Type: %@", @"AES"]]
                            break;
                        case CAPTURE_ENCRYPT_TYPE_TDES:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encrypt Type: %@", @"TDES"]]
                            break;
                        case CAPTURE_ENCRYPT_TYPE_NO_ENCRYPTION:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encrypt Type: %@", @"NONE"]]
                            break;
                        default:
                            [self appendMessageToResults:[NSString stringWithFormat:@"Encrypt 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]];
    if (cardData.unencryptedTags != nil) [self appendMessageToResults:[NSString stringWithFormat:@"
Unencrypted Tags: %@", cardData.unencryptedTags.description]];
    if (cardData.encryptedTags != nil) [self appendMessageToResults:[NSString stringWithFormat:@"
Encrypted Tags: %@", cardData.encryptedTags.description]];
    if (cardData.maskedTags != nil) [self appendMessageToResults:[NSString stringWithFormat:@"
Masked Tags: %@", cardData.maskedTags.description]];

    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 protocol delegate `plugStatusChange:()` to automatically attempt connection attempt.

```

- (void) plugStatusChange: (BOOL)deviceInserted{
    if (deviceInserted) {
        [self appendMessageToResults: @"device Attached. Attempting connection"];
        [[IDT_UniPayI_V sharedController] device_connectToAudioReader];
    }
    else{
        [self appendMessageToResults: @"device removed."];
    }
}

```

- Implement protocol delegate `emvTransactionData:()` to report EMV transaction results

```

- (void) emvTransactionData: (IDTEMVData*)emvData errorCode: (int)error{
    NSLog(@"EMV_RESULT_CODE_V2_response = %2X",error);

    [self appendMessageToResults:[NSString stringWithFormat:@"EMV_RESULT_CODE_V2_response = %2X",error]];
}

```

```

if (emvData == nil) {
    [self appendMessageToResults:[NSString stringWithFormat:@"EMV TRANSACTION ERROR. Refer to
    EMV_RESULT_CODE_V2_response = 0x%2X",error]];
    return;
}

if (emvData.resultCodeV2 == EMV_RESULT_CODE_V2_GO_ONLINE) {
    [self appendMessageToResults:@"ONLINE REQUEST"];
}
if (emvData.resultCodeV2 == EMV_RESULT_CODE_V2_APPROVED || emvData.resultCodeV2 ==
    EMV_RESULT_CODE_V2_APPROVED_OFFLINE ) {
    [self appendMessageToResults:@"APPROVED"];
}
if (emvData.resultCodeV2 == EMV_RESULT_CODE_V2_MSR_SUCCESS) {
    [self appendMessageToResults:@"MSR Data Captured"];
}

if (emvData.cardType == 0) {
    [self appendMessageToResults:@"CONTACT"];
}
if (emvData.cardType == 1) {
    [self appendMessageToResults:@"CONTACTLESS"];
}

if (emvData.unencryptedTags != nil) [self appendMessageToResults:[NSString stringWithFormat:@"
Unencrypted Tags: %@", emvData.unencryptedTags.description]];
if (emvData.encryptedTags != nil) [self appendMessageToResults:[NSString stringWithFormat:@"Encrypted
Tags: %@", emvData.encryptedTags.description]];
if (emvData.maskedTags != nil) [self appendMessageToResults:[NSString stringWithFormat:@"Masked Tags:
%@", emvData.maskedTags.description]];
}

```

- Implement protocol delegate `lcdDisplay:()` to receive LCD messages, and automatically select 1st menu item/language when presented with choices. Normal operation would require a choice be made by card holder.

```

- (void) lcdDisplay:(int)mode lines:(NSArray*)lines{
    NSMutableString* str = [NSMutableString new];
    if (lines != nil) {
        for (NSString* s in lines) {
            [str appendString:s];
            [str appendString:@"\n"];
        }
    }

    switch (mode) {
        case 0x10:
            //clear screen
            lcdTextView.text = @"";
            break;
        case 0x03:
            lcdTextView.text = str;
            break;
        case 0x01:
        case 0x02:
        case 0x08:{
            [[IDT_UniPayI_V sharedController] emv_callbackResponseLCD:mode selection:(unsigned
            char)1];
        }
        break;
        default:
            break;
    }
}

```

- Implement the button press methods

```

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

- (IBAction) startMSR_CTLS:(id)sender{
    logTextView.text = @"";
    RETURN_CODE rt = [[IDT_UniPayI_V sharedController] msr_startMSRSwipe];
    if (RETURN_CODE_DO_SUCCESS == rt){
        [self appendMessageToResults:@"EnableMSR: OK."];
    }
}

```

```

    }
    -(IBAction) startEMV:(id)sender{
        logTextView.text = @"";
        RETURN_CODE rt = [[IDT_UniPayI_V sharedController] emv_startTransaction:1.00 amtOther:0
            type:0 timeout:60 tags:nil forceOnline:false fallback:true];
        if (RETURN_CODE_DO_SUCCESS == rt)
        {
            [self appendMessageToResults: @"Start Transaction Command Accepted"];
        }
    }
    -(IBAction) completeEMV:(id)sender{
        logTextView.text = @"";
        RETURN_CODE rt = [[IDT_UniPayI_V sharedController] emv_completeOnlineEMVTransaction:true
            hostResponseTags:[IDTUtility hexToData:@"8A023030"] ];
        if (RETURN_CODE_DO_SUCCESS == rt)
        {
            [self appendMessageToResults: @"Complete Transaction Command Accepted"];
        }
    }
    -(IBAction) cancelTransaction:(id)sender{
        logTextView.text = @"";
        RETURN_CODE rt = [[IDT_UniPayI_V sharedController] msr_cancelMSRSwipe];
        if (RETURN_CODE_DO_SUCCESS == rt){
            [self appendMessageToResults:@"CancelMSR: OK."];
        }
    }
}

```



## Chapter 7

# Core Implementation UniPay 1.5: Swift

IDTech Framework includes class libraries to interface with the UniPay 1.5. This guide assume a fair understanding of Xcode 7.0+ and general Apple iOS programming knowledge.

### 7.1 Integrating with IDTech Framework

- [Import the Necessary Framework/Libraries](#)
- [Create a Bridging Header File](#)
- [Add Import Statement to the Bridging Header File](#)
- [Amend the View Controller Interface](#)
- [Implement Optional Delegate Protocols](#)
- [Allocate/Initialize IDT\\_UniPayI\\_V Object](#)
- [Sample Project Tutorial](#)

### 7.2 Import the Necessary Framework/Libraries

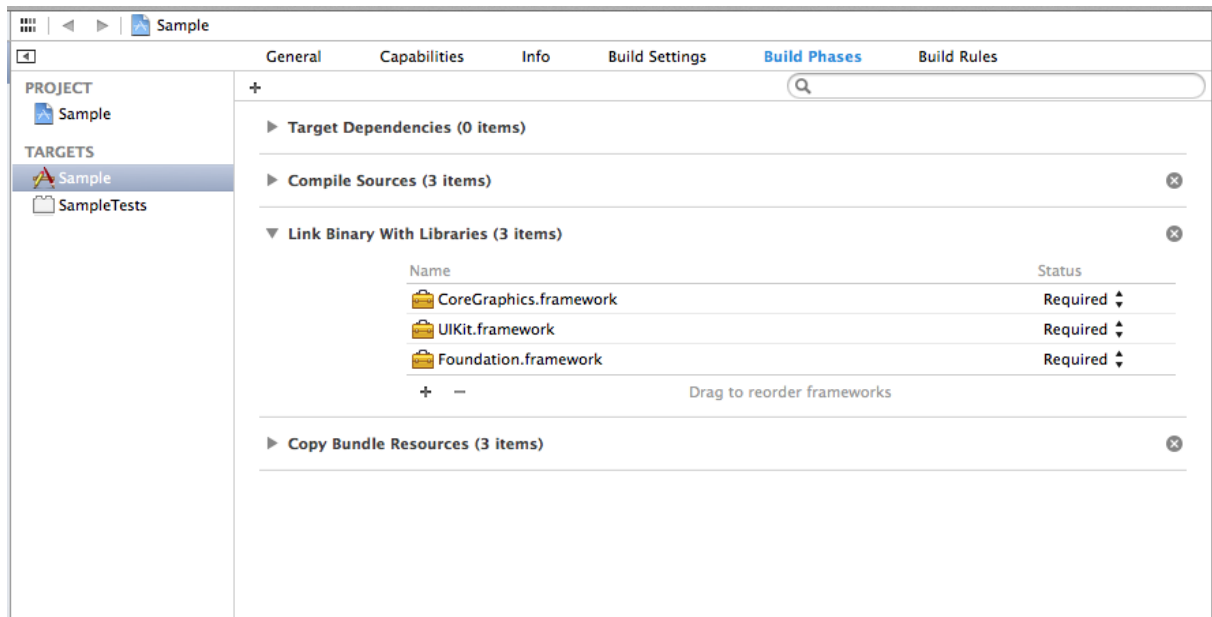
Communicating with IDTech Devices requires the following framework/libraries to be imported into the project:

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

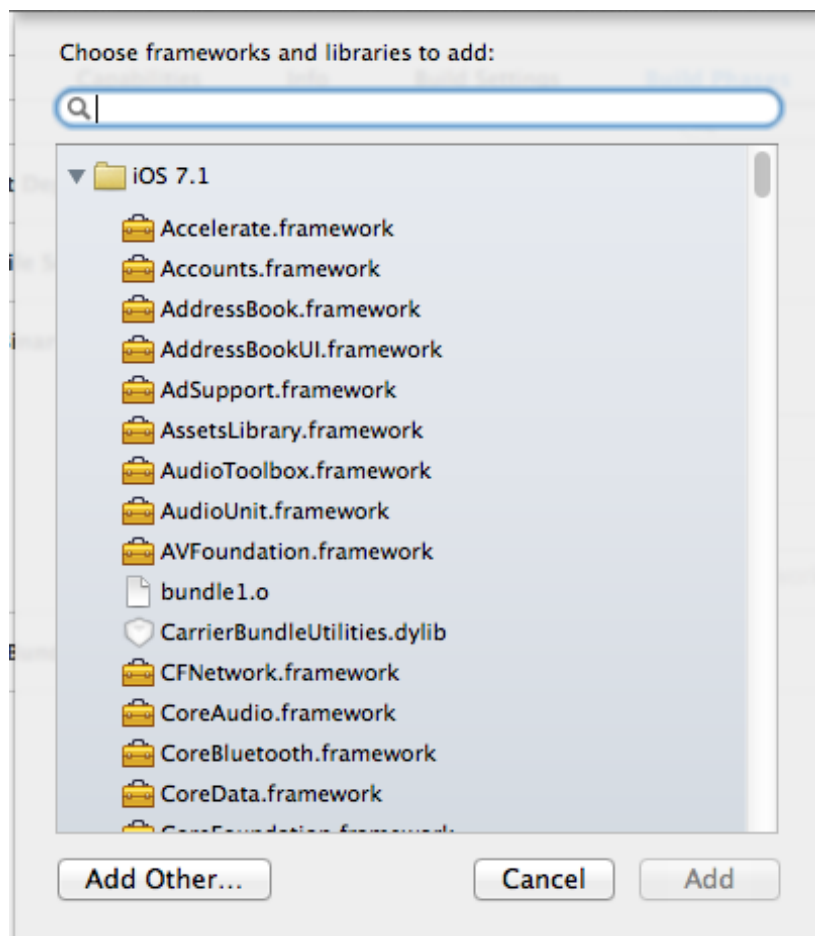
Also, import the following resource bundle:

- IDTech.bundle

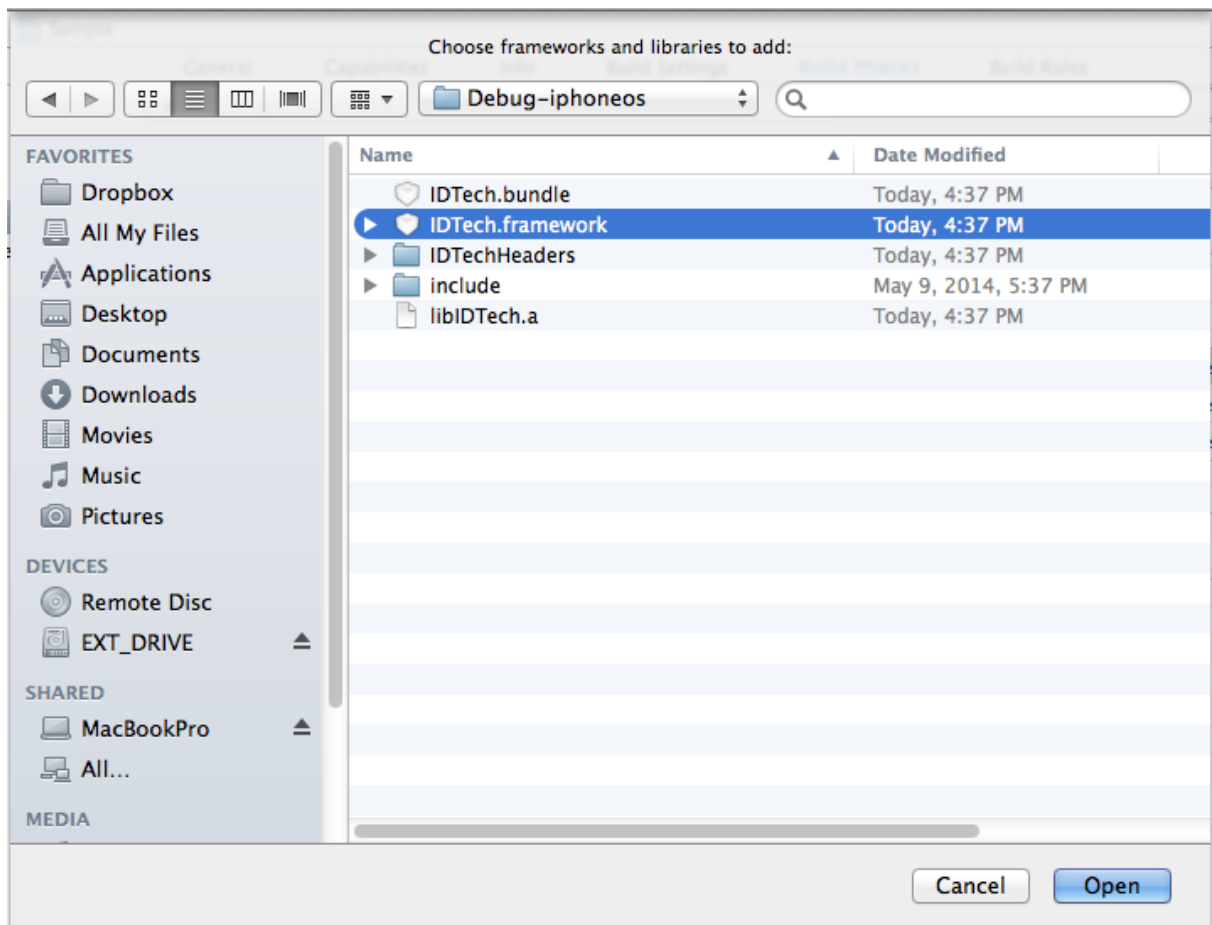
Under Build Phases, select Link Binary With Libraries and click the Add (+) button



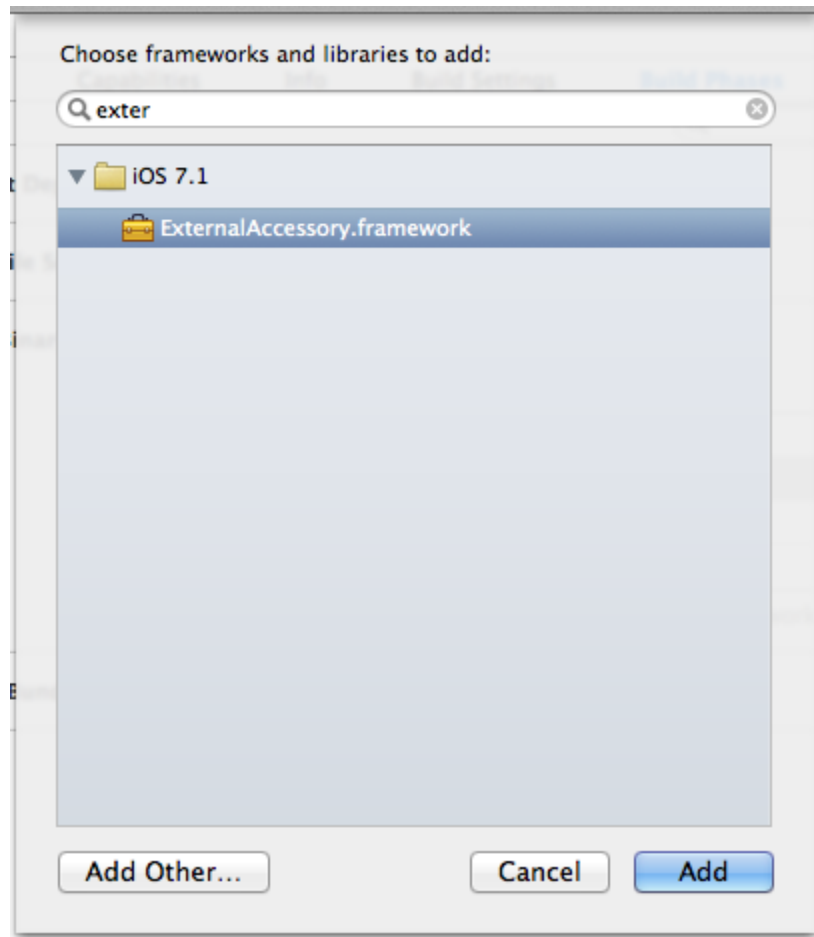
On the Choose Frameworks screen, click "Add Other" in the lower left



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



Link the ExternalAccessory framework. On the Choose Frameworks screen, type "exter" into the search bar, select ExternalAccessory.framework and click "Add"

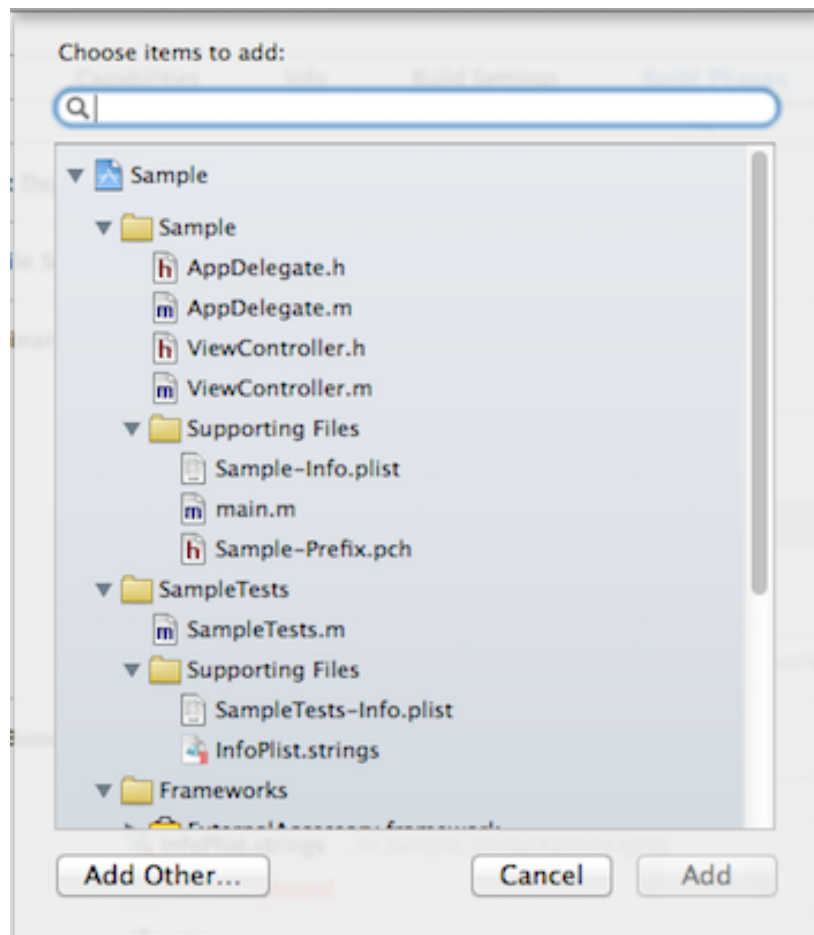


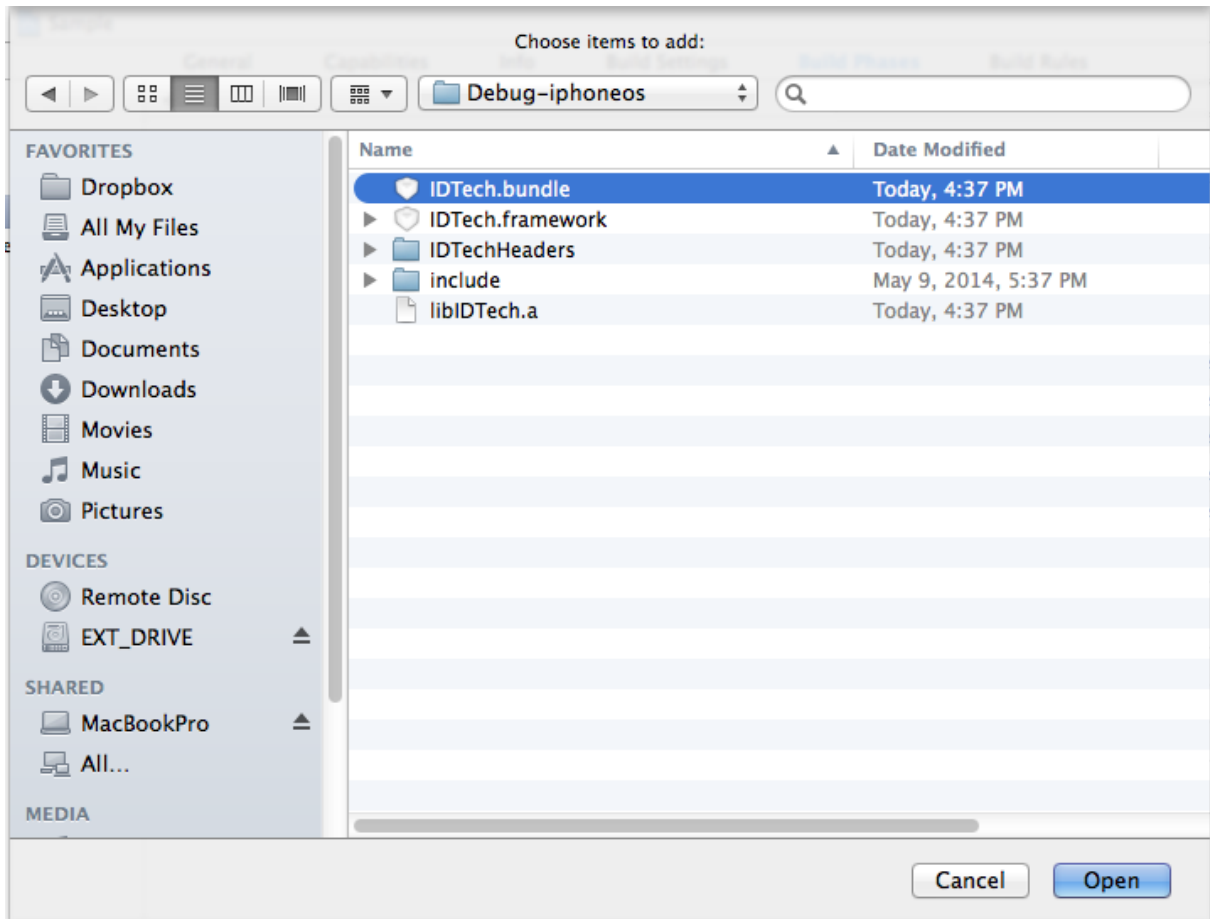
Repeat process for MediaPlayer.framework, AVFoundation.framework, AudioToolbox.framework, and CFNetwork.framework

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

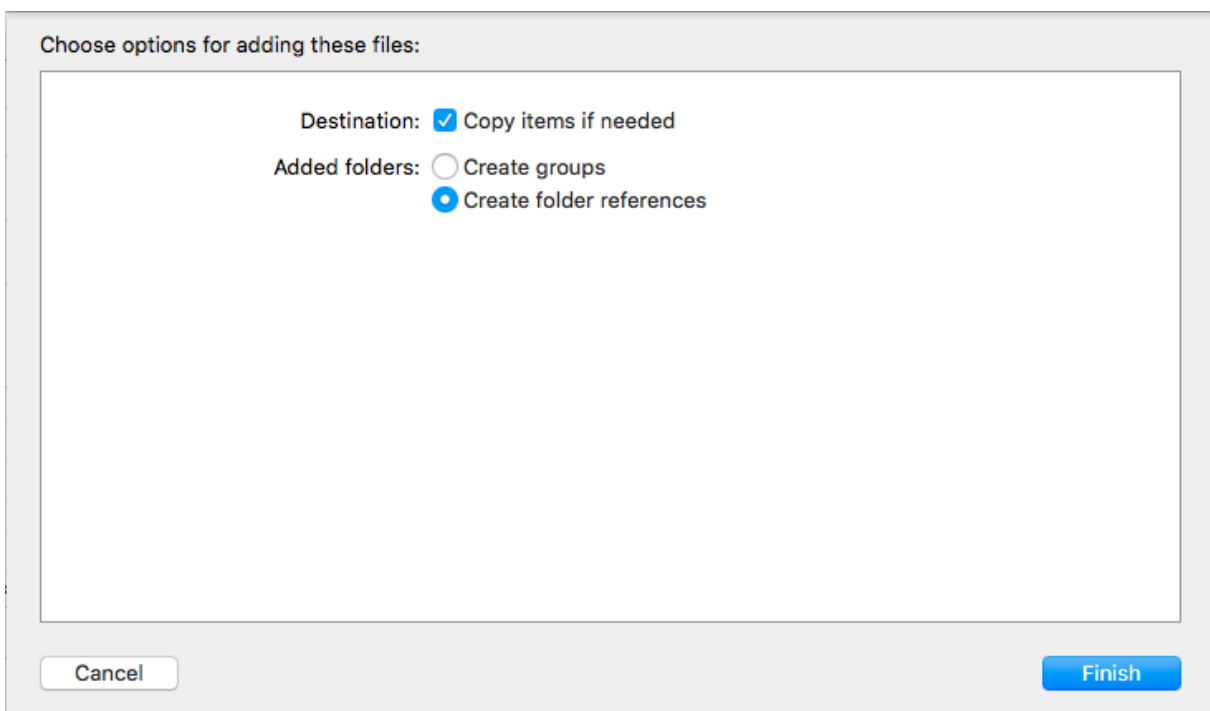
### ▼ Copy Bundle Resources (3 items)

- ☐ LaunchScreen.storyboard
  - ☒ Assets.xcassets ...in UniPay\_Swift\_Tutorial
  - ☐ Main.storyboard
- + —

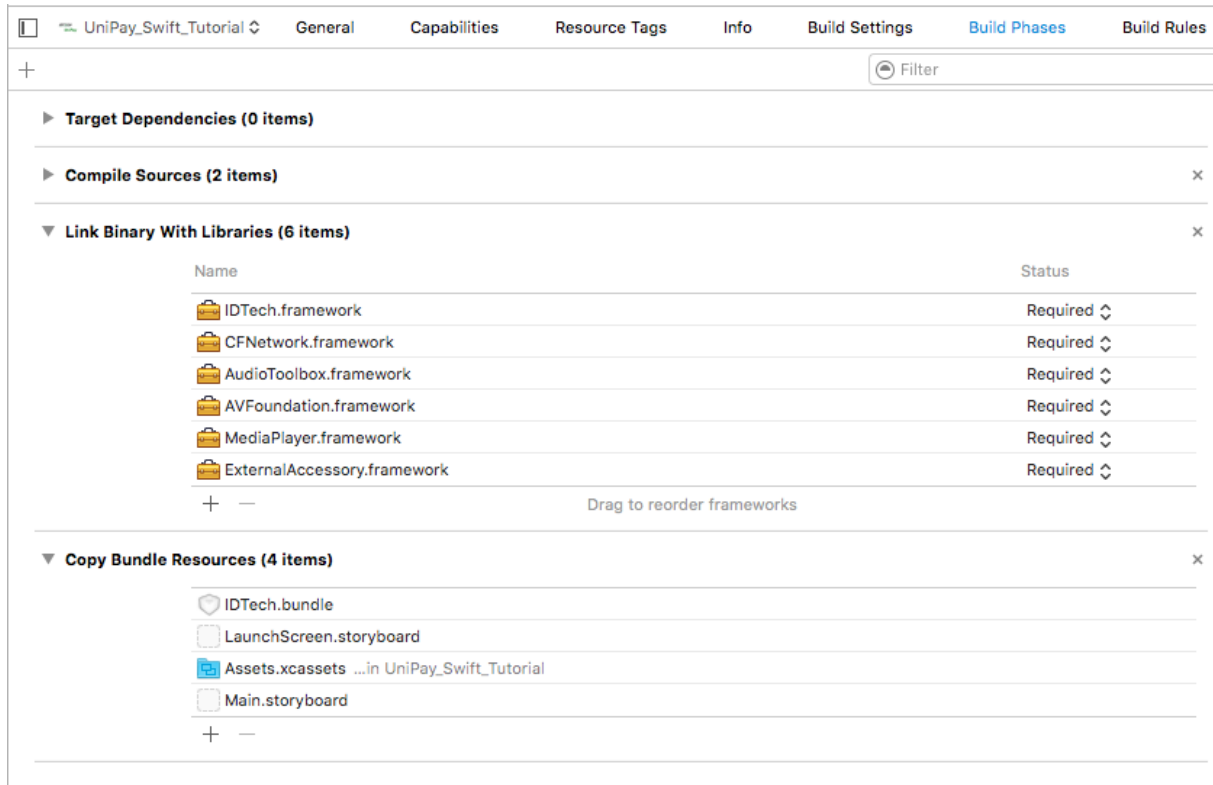




On the following option screen, leave the default selections and click "Finish"



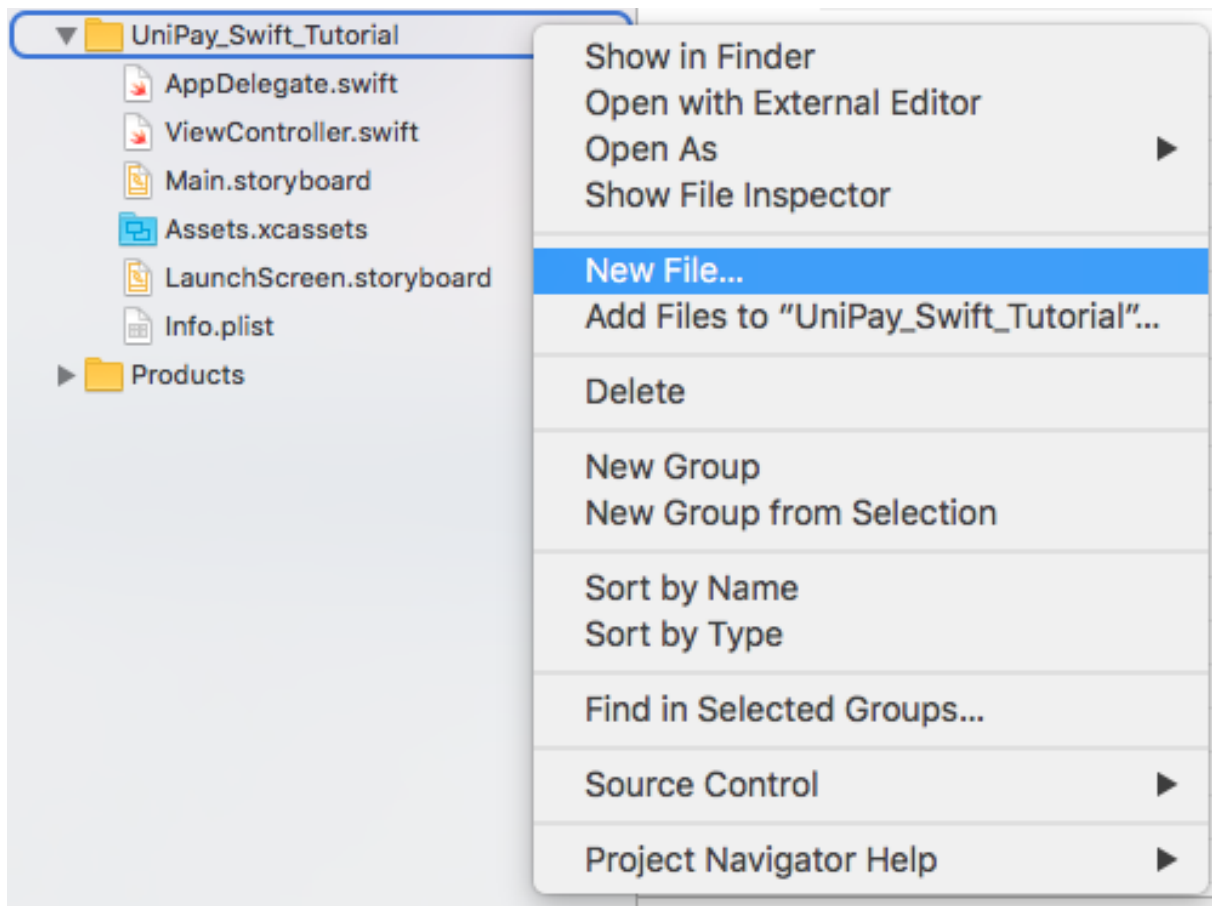
The Build Phases should now include the required frameworks/libraries for the UniPay 1.5



## 7.3 Create a Bridging Header File

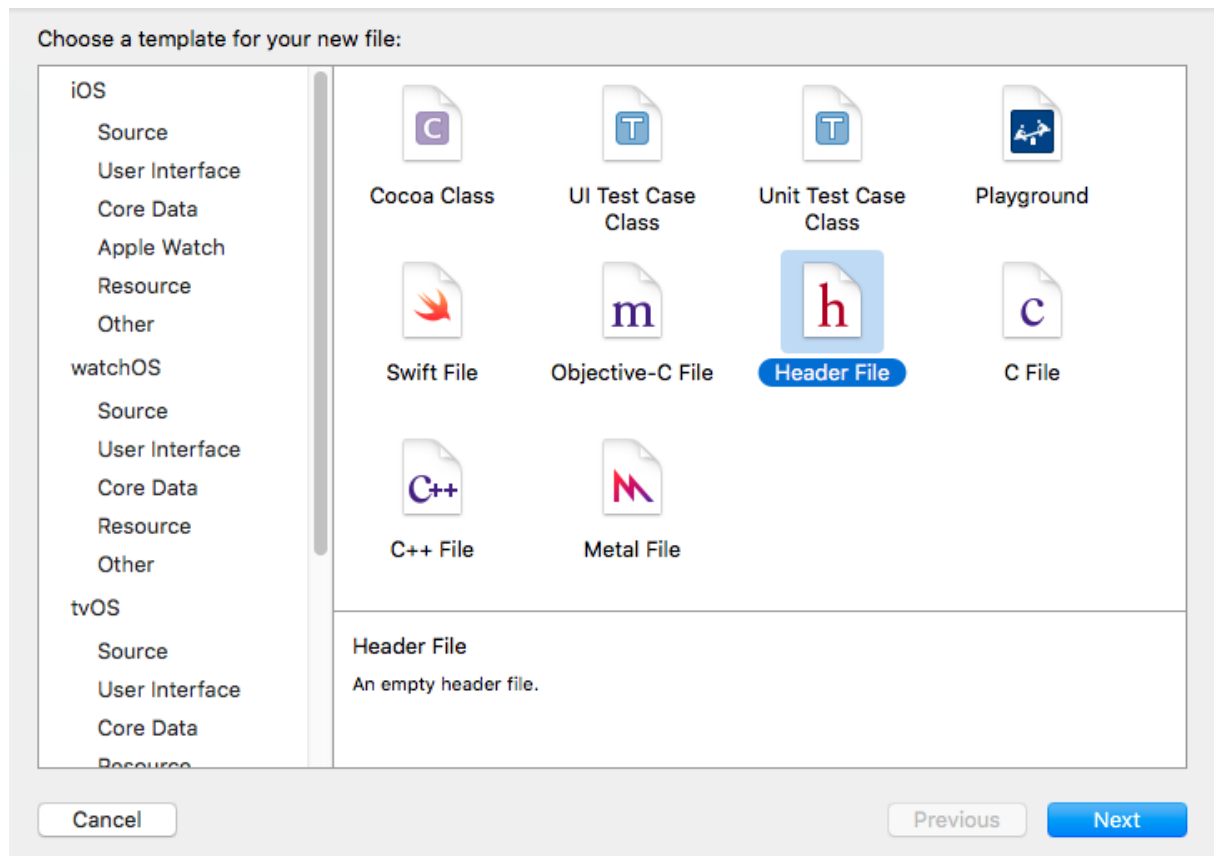
Using the UniPay SDK with Swift requires a bridging header to allow the application to have a mixed-language codebase

First, you will need to create a header file. Right click on your project directory and click "New File...".

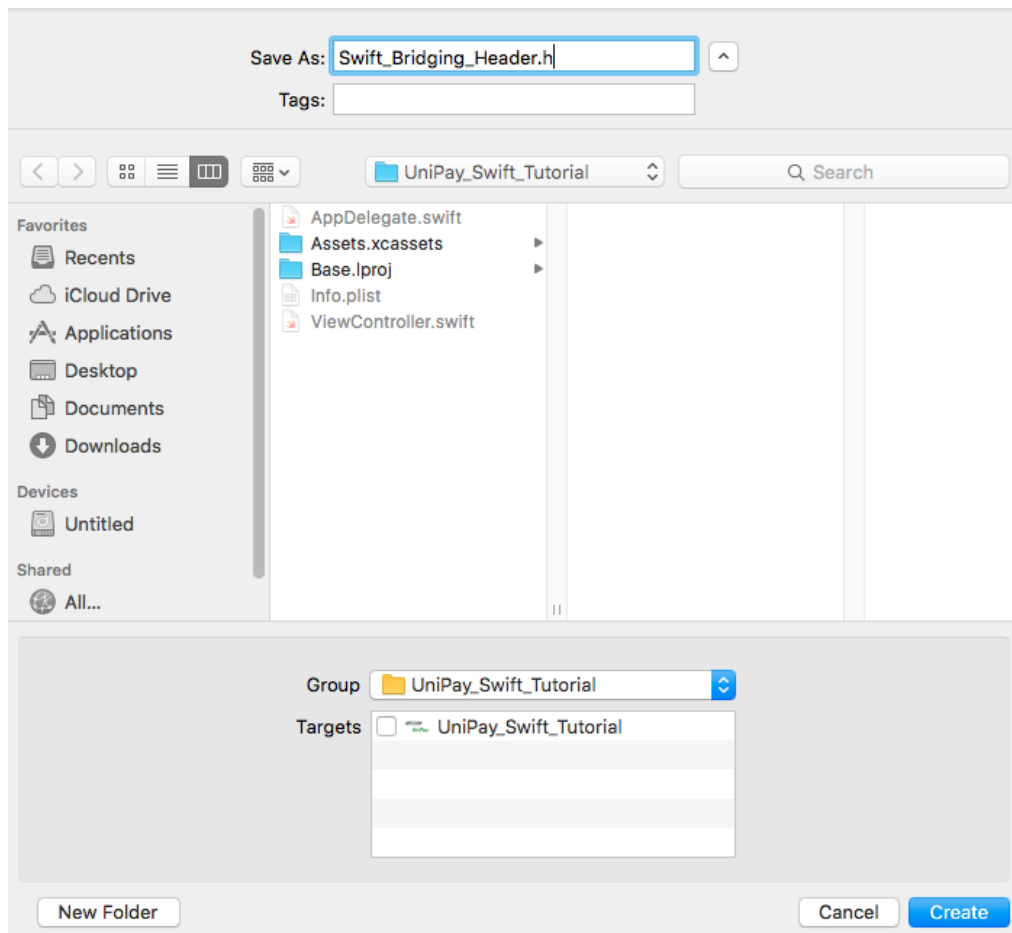


On the file creation screen, select Header File and click "Next"

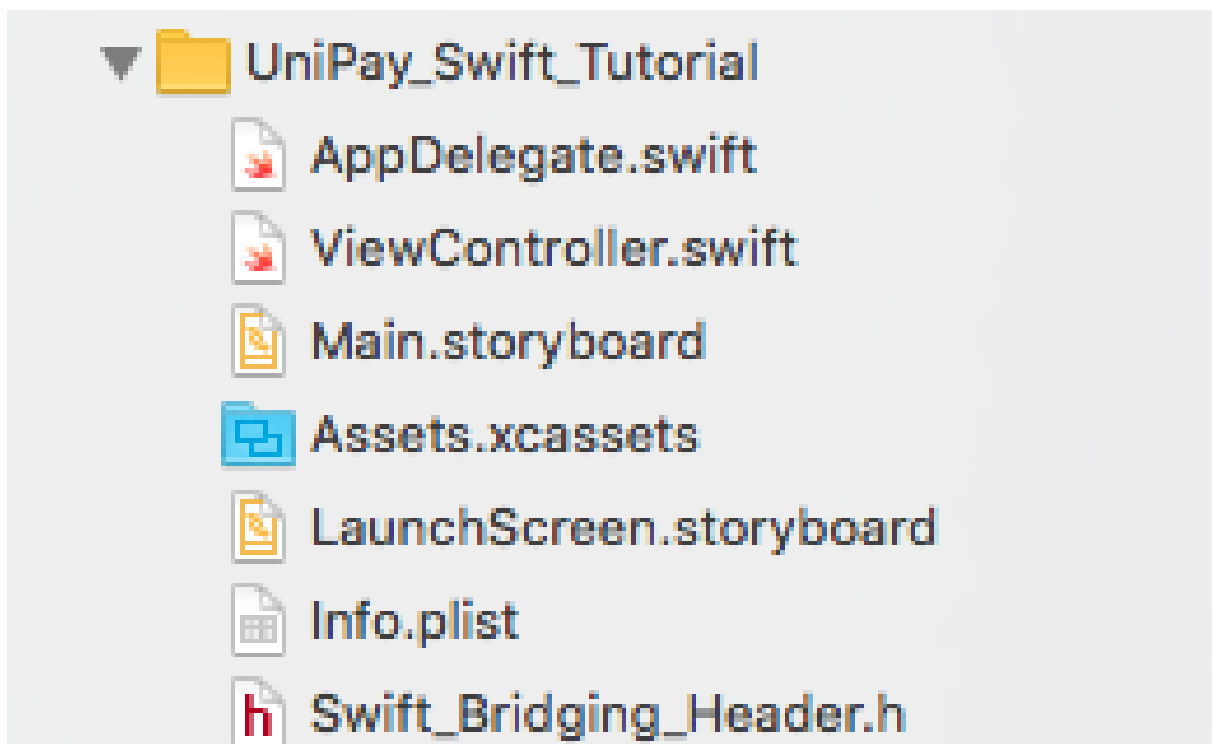




On the following screen, choose a name for your header file and click "Create"



The bridging header file is now created

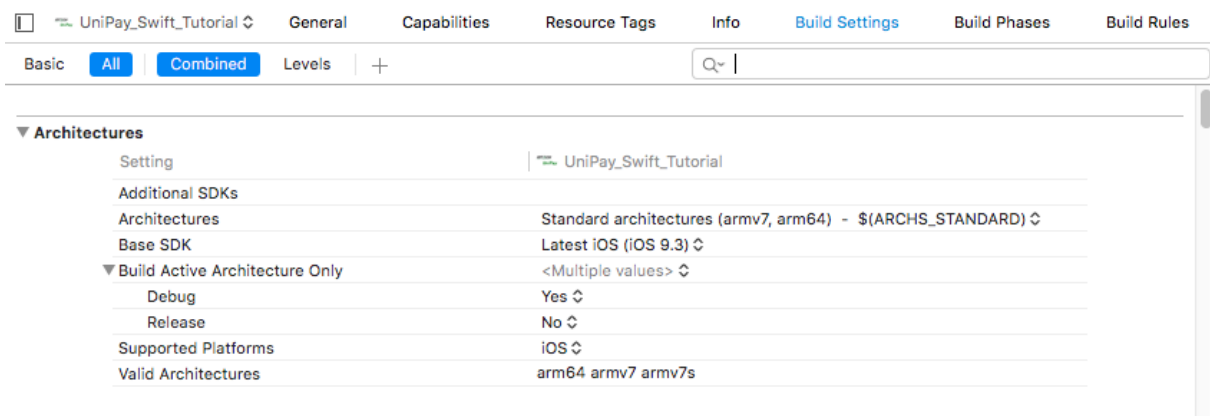


The header file should look similar to the following:

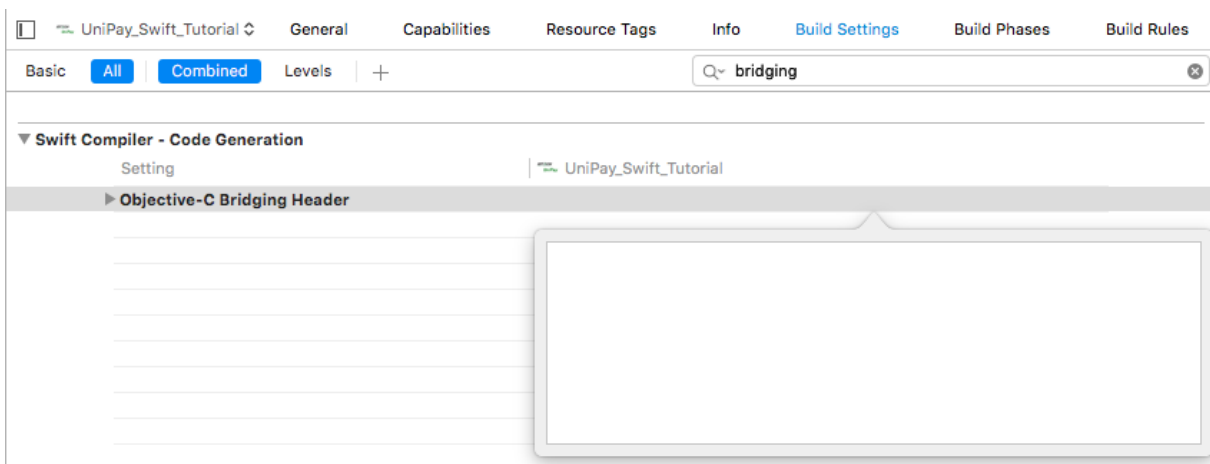
```
#ifndef Swift_Bridging_Header_h
#define Swift_Bridging_Header_h

#endif
```

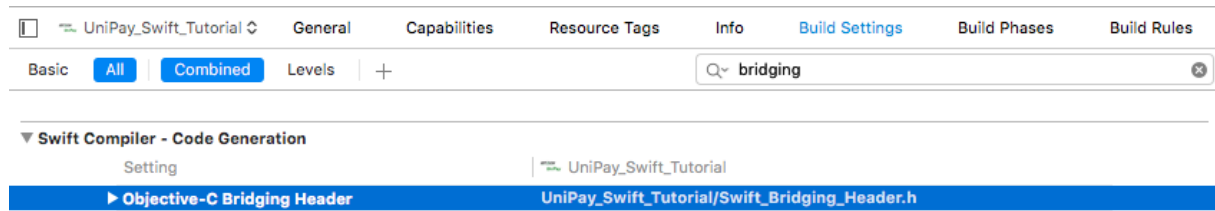
Next, you will need to link your header file to the bridging header setting. Navigate to Build Settings within Xcode



Type "bridging" into the search bar and double click the Objective-C Bridging Header setting to get an input box



Type the relative path to your project directory and the name of the bridging header file and then click enter to save the setting



## 7.4 Add Import Statement to the Bridging Header File

Inside of the newly created bridging header file, add the following import statement:

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

The file should now look similar to the following:

```
#ifndef Swift_Bridging_Header_h
#define Swift_Bridging_Header_h

#import <IDTech/IDTech.h>

#endif
```

## 7.5 Amend the View Controller Interface

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

```
class ViewController: UIViewController, IDT_UniPayI_V_Delegate
```

## 7.6 Implement Optional Delegate Protocols

Implement any/all of the optional delegate protocols used to receive data from [IDT\\_UniPayI\\_V\\_Delegate](#):

```
func deviceConnected()
func deviceDisconnected()
func plugStatusChange(deviceInserted: Bool)
func dataInOutMonitor(data: NSData!, incoming isIncoming: Bool)
func swipeMSRData(cardData: IDTMSRData!)
```

```
func deviceMessage(message: NSString!)  
  
func lcdDisplay(mode: Int32, lines: [AnyObject]!)  
  
func emvTransactionData(emvData: IDTEMVData!, errorCode error: Int32)
```

## 7.7 Allocate/Initialize IDT\_UniPayI\_V Object

A Singleton instance has been established in the [IDT\\_UniPayI\\_V](#) class. To utilize the delegate protocols, best practices would be to initialize the connection by setting the delegate with the singleton instance.

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    // Do any additional setup after loading the view, typically from a nib.  
  
    IDT_UniPayI_V.sharedController().delegate = self  
}
```

## 7.8 Sample Project Tutorial

Using Xcode 7.3.1, we will create a sample project that will interface with the UniPay 1.5 and will perform the following activities:

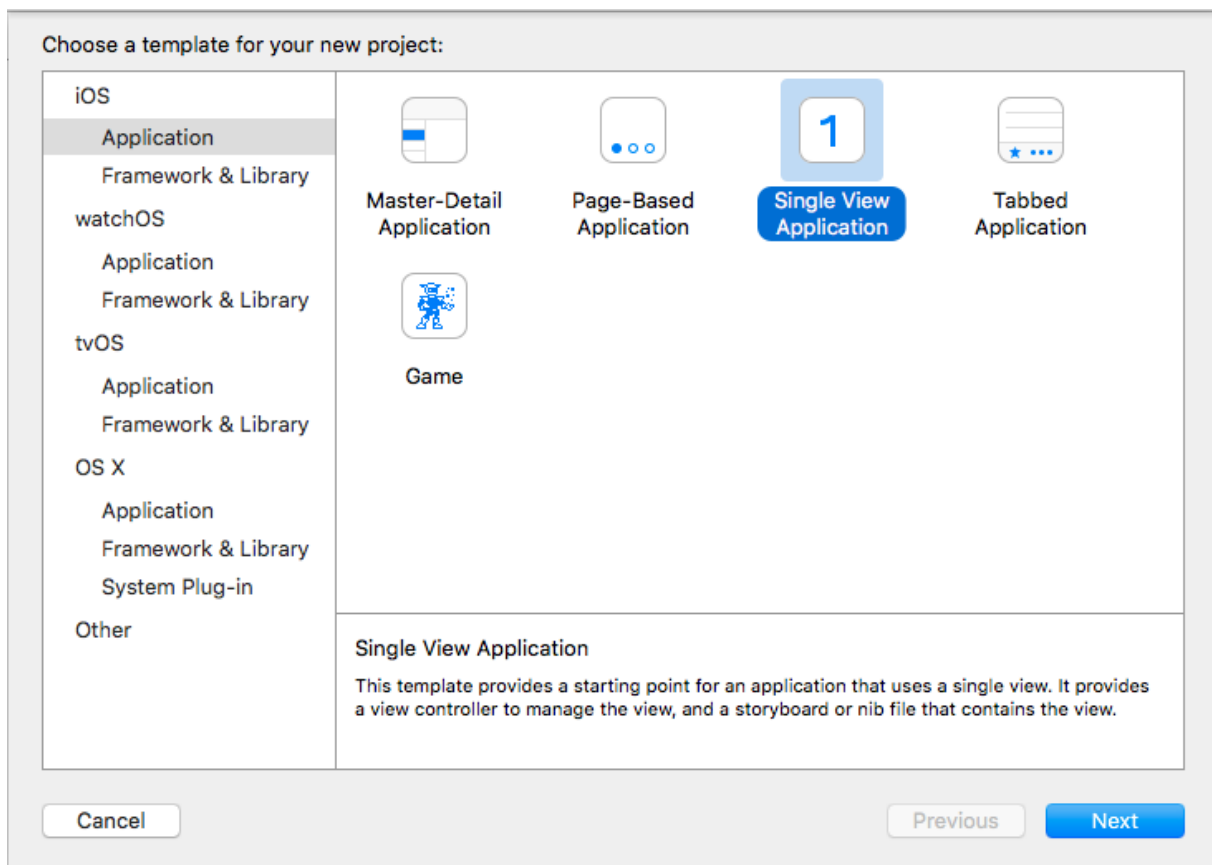
- Auto-Connect and display connection status
- Get Device Firmware
- Start/Stop Transaction Request for MSR
- Start/Complete/Cancel EMV Transaction
- Show LCD Display for EMV transaction
- Clear the Displays
- Automatically select first AID or first Language if prompted

Protocol Delegates:

- Delegate to receive card swipes
- Delegate to detect headphone plug changes
- Delegate to detect device connected
- Delegate to detect device disconnected
- Delegate to receive EMV tag data

### 7.8.1 Step 1: Create New Project

Create a new Single View Application in Xcode



Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

☐ Use Core Data

☐ Include Unit Tests

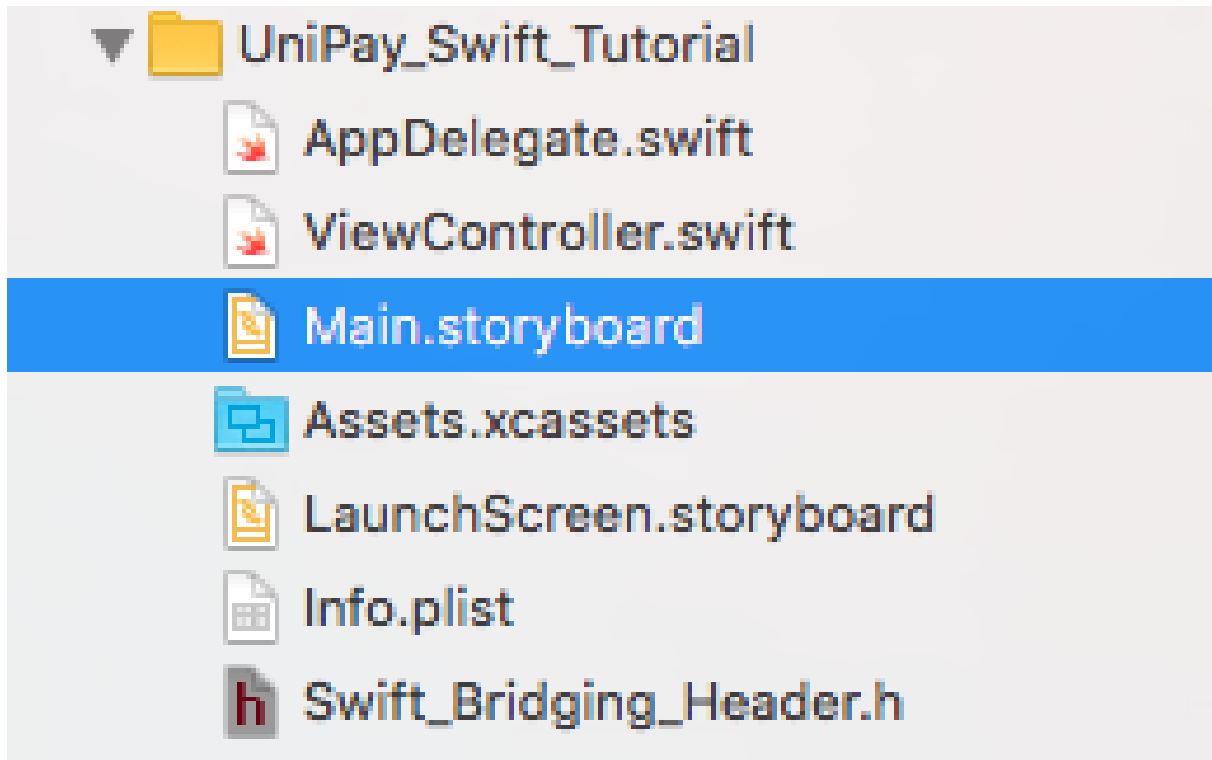
☐ Include UI Tests

## 7.8.2 Step 2: Import Frameworks

[Import the Necessary Framework/Libraries](#)

## 7.8.3 Step 3: Design Interface

Design the User Interface by editing the Main.storyboard file



Open your storyboard and add items so it contains the following buttons/fields:

- Add a navigation bar at the top to display the application's name and hold a bar button item
- Add a label at the top that will signify the connection/disconnection status.
- Add text views to communicate data from the UniPay 1.5 and for EMV LCD display information. Remove the Editable and Selectable behaviors 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 MSR
  - Start ICC EMV
  - Complete ICC EMV
  - Cancel Transaction
  - Clear the Displayed Information (bar button item on the navigation bar)  
(Add constraints accordingly so the layout maps to the intended screen size)

#### 7.8.4 Step 4: Configure the Bridging Header and View Controller Files

Create the bridging header file by performing the following:

- [Create a Bridging Header File](#)

In the bridging header file, perform the following:

- [Add Import Statement to the Bridging Header](#)



In the view controller file, perform the following:

- [Amend the View Controller Interface](#)
- Create an IBOutlet for the two 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 6 IBAction for the buttons, and link them to the "Touch Up Inside" event on the storyboard buttons

```
import UIKit

class ViewController: UIViewController, IDT_UniPayI_V_Delegate {
    @IBOutlet weak var connectionStatus: UILabel!
    @IBOutlet weak var lcdTextView: UITextView!
    @IBOutlet weak var logTextView: UITextView!

    @IBAction func getFirmware(sender: UIButton) {}
    @IBAction func startMSR(sender: UIButton) {}
    @IBAction func startICCEMV(sender: UIButton) {}
    @IBAction func completeICCEMV(sender: UIButton) {}
    @IBAction func cancelTransaction(sender: UIButton) {}
    @IBAction func clearOutputs(sender: UIBarButtonItem) {}
}
```

### Storyboard Source Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB" version="3.0" toolsVersion="10117"
    systemVersion="15F34" targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES"
    useTraitCollections="YES" initialViewController="BYZ-38-t0r">
    <dependencies>
        <deployment identifier="iOS"/>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="10085"/>
        <capability name="Constraints to layout margins" minToolsVersion="6.0"/>
    </dependencies>
    <scenes>
        <!--View Controller-->
        <scene sceneID="tne-Qt-ifu">
            <objects>
                <viewController id="BYZ-38-t0r" customClass="ViewController" customModule="
UniPay_Swift_Tutorial" customModuleProvider="target" sceneMemberID="viewController">
                    <layoutGuides>
                        <viewControllerLayoutGuide type="top" id="y3c-jy-aDJ"/>
                        <viewControllerLayoutGuide type="bottom" id="wfy-db-euE"/>
                    </layoutGuides>
                    <view key="view" contentMode="scaleToFill" id="8bC-Xf-vdC">
                        <rect key="frame" x="0.0" y="0.0" width="600" height="600"/>
                        <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
                        <subviews>
                            <navigationBar contentMode="scaleToFill"
                                translatesAutoresizingMaskIntoConstraints="NO" id="SOj-lk-9zM">
                                <rect key="frame" x="0.0" y="28" width="600" height="44"/>
                                <items>
                                    <navigationItem title="UniPay Demo (Swift)" id="u9K-Bw-PuF">
                                        <barButtonItem key="rightBarButtonItem" systemItem="trash" id="
VYJ-YX-Lrd">
                                            <connections>
                                                <action selector="clearOutputs:" destination="BYZ-38-t0r"
id="pCm-WP-cts"/>
                                            </connections>
                                        </barButtonItem>
                                    </navigationItem>
                                </items>
                            </navigationBar>
                            <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
                                horizontalHuggingPriority="251" verticalHuggingPriority="251" text="Disconnected" textAlignment="center" lineBreak
                                tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
                                translatesAutoresizingMaskIntoConstraints="NO" id="xu3-cC-StE">
                                    <rect key="frame" x="0.0" y="72" width="600" height="21"/>
                                    <color key="backgroundColor" red="1" green="0.0" blue="0.0" alpha="1"
colorSpace="calibratedRGB"/>
                                    <fontDescription key="fontDescription" type="system" pointSize="17"/>
                                    <color key="textColor" cocoaTouchSystemColor="darkTextColor"/>
                                    <nil key="highlightedColor"/>
                                </label>
                        </subviews>
                    </view>
                </viewController>
            </objects>
        </scene>
    </scenes>
</document>
```

```

        <view contentMode="scaleToFill" translatesAutoresizingMaskIntoConstraints="NO"
id="QAv-yD-BWh">
            <rect key="frame" x="0.0" y="93" width="600" height="114"/>
            <subviews>
                <button opaque="NO" contentMode="scaleToFill"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="mid
translatesAutoresizingMaskIntoConstraints="NO" id="RFL-d7-zWi">
                    <rect key="frame" x="8" y="8" width="112" height="30"/>
                    <constraints>
                        <constraint firstAttribute="height" constant="30" id="
8yF-f4-31b"/>
                        <constraint firstAttribute="width" constant="112" id="
UDu-xZ-Oo2"/>
                    </constraints>
                    <state key="normal" title="Get Firmware"/>
                    <connections>
                        <action selector="getFirmware:" destination="BYZ-38-t0r"
eventType="touchUpInside" id="15Y-tc-hG1"/>
                    </connections>
                </button>
                <button opaque="NO" contentMode="scaleToFill"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="mid
translatesAutoresizingMaskIntoConstraints="NO" id="PIJ-Kn-LEk">
                    <rect key="frame" x="502" y="8" width="90" height="30"/>
                    <constraints>
                        <constraint firstAttribute="width" constant="90" id="I1I-3z-cT6
"/>
                        <constraint firstAttribute="height" constant="30" id="
RYN-YH-3Ew"/>
                    </constraints>
                    <state key="normal" title="Start MSR"/>
                    <connections>
                        <action selector="startMSR:" destination="BYZ-38-t0r" eventType
="touchUpInside" id="Uvq-vw-5qL"/>
                    </connections>
                </button>
                <button opaque="NO" contentMode="scaleToFill"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="mid
translatesAutoresizingMaskIntoConstraints="NO" id="0dY-NY-EXD">
                    <rect key="frame" x="441" y="46" width="151" height="30"/>
                    <constraints>
                        <constraint firstAttribute="height" constant="30" id="
16Z-8F-ycW"/>
                        <constraint firstAttribute="width" constant="151" id="
3nw-Xz-PNG"/>
                    </constraints>
                    <state key="normal" title="Complete ICC EMV"/>
                    <connections>
                        <action selector="completeICCEMV:" destination="BYZ-38-t0r"
eventType="touchUpInside" id="n85-tf-w3z"/>
                    </connections>
                </button>
                <button opaque="NO" contentMode="scaleToFill"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="mid
translatesAutoresizingMaskIntoConstraints="NO" id="Cbp-oM-qBO">
                    <rect key="frame" x="234" y="84" width="132" height="30"/>
                    <constraints>
                        <constraint firstAttribute="height" constant="30" id="
DUR-tL-q39"/>
                        <constraint firstAttribute="width" constant="132" id="
Ubg-Xu-IJg"/>
                    </constraints>
                    <state key="normal" title="Cancel Transaction"/>
                    <connections>
                        <action selector="cancelTransaction:" destination="BYZ-38-t0r"
eventType="touchUpInside" id="8Lj-Gd-th0"/>
                    </connections>
                </button>
                <button opaque="NO" contentMode="scaleToFill"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="mid
translatesAutoresizingMaskIntoConstraints="NO" id="ndD-aJ-DMC">
                    <rect key="frame" x="8" y="46" width="118" height="30"/>
                    <constraints>
                        <constraint firstAttribute="height" constant="30" id="
fE9-68-i3S"/>
                        <constraint firstAttribute="width" constant="118" id="
p5K-sa-b7N"/>
                    </constraints>
                    <state key="normal" title="Start ICC EMV"/>
                    <connections>
                        <action selector="startICCEMV:" destination="BYZ-38-t0r"
eventType="touchUpInside" id="Kas-aF-Nvx"/>
                    </connections>
                </button>
            </subviews>
            <color key="backgroundColor" white="1" alpha="1" colorSpace="
calibratedWhite"/>

```

```

        <constraints>
            <constraint firstAttribute="trailing" secondItem="PIJ-Kn-LEk"
secondAttribute="trailing" constant="8" id="2CN-Uw-z4v"/>
            <constraint firstItem="RFL-d7-zWi" firstAttribute="top" secondItem="
QAv-yD-BWh" secondAttribute="top" constant="8" id="45Y-Fc-f90"/>
            <constraint firstItem="0dY-NY-EXD" firstAttribute="top" secondItem="
PIJ-Kn-LEk" secondAttribute="bottom" constant="8" id="5Yr-a7-vbt"/>
            <constraint firstItem="Cbp-oM-qBO" firstAttribute="top" secondItem="
QAv-yD-BWh" secondAttribute="top" constant="84" id="IXa-51-4nR"/>
            <constraint firstItem="RFL-d7-zWi" firstAttribute="leading" secondItem="
QAv-yD-BWh" secondAttribute="leading" constant="8" id="KM6-d4-Jcf"/>
            <constraint firstItem="PIJ-Kn-LEk" firstAttribute="top" secondItem="
QAv-yD-BWh" secondAttribute="top" constant="8" id="PHS-zD-pGC"/>
            <constraint firstItem="ndD-aJ-DMC" firstAttribute="top" secondItem="
RFL-d7-zWi" secondAttribute="bottom" constant="8" id="avp-CL-dRU"/>
            <constraint firstItem="Cbp-oM-qBO" firstAttribute="centerX" secondItem="
QAv-yD-BWh" secondAttribute="centerX" id="1LD-Qr-wlp"/>
            <constraint firstAttribute="trailing" secondItem="0dY-NY-EXD"
secondAttribute="trailing" constant="8" id="1jq-ay-cry"/>
            <constraint firstItem="ndD-aJ-DMC" firstAttribute="leading" secondItem="
QAv-yD-BWh" secondAttribute="leading" constant="8" id="nP6-4i-DHM"/>
            <constraint firstAttribute="height" constant="114" id="phB-Nt-IAS"/>
        </constraints>
    </view>
    <view contentMode="scaleToFill" translatesAutoresizingMaskIntoConstraints="NO"
id="6lI-3h-wdU">
        <rect key="frame" x="0.0" y="215" width="600" height="385"/>
        <subviews>
            <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
scaleToFill" editable="NO" textAlignment="natural" selectable="NO" translatesAutoresizingMaskIntoConstraints=
"NO" id="mqx-Yb-8Le">
                <rect key="frame" x="8" y="192" width="584" height="184"/>
                <color key="backgroundColor" white="0.6666666666999997" alpha="
0.14835831930000001" colorSpace="calibratedWhite"/>
                <fontDescription key="fontDescription" type="system" pointSize="14"
/>
                <textInputTraits key="textInputTraits" autocapitalizationType="
sentences"/>
            </textView>
            <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
scaleToFill" editable="NO" textAlignment="natural" selectable="NO" translatesAutoresizingMaskIntoConstraints=
"NO" id="0il-CG-C0J">
                <rect key="frame" x="8" y="0.0" width="584" height="184"/>
                <color key="backgroundColor" white="0.6666666666999997" alpha="
0.14835831930000001" colorSpace="calibratedWhite"/>
                <fontDescription key="fontDescription" type="system" pointSize="14"
/>
                <textInputTraits key="textInputTraits" autocapitalizationType="
sentences"/>
            </textView>
        </subviews>
        <color key="backgroundColor" white="1" alpha="1" colorSpace="
calibratedWhite"/>
        <constraints>
            <constraint firstAttribute="trailing" secondItem="mqx-Yb-8Le"
secondAttribute="trailing" constant="8" id="6Gc-50-OI5"/>
            <constraint firstItem="mqx-Yb-8Le" firstAttribute="top" secondItem="
0il-CG-C0J" secondAttribute="bottom" constant="8" id="8Qm-b4-009"/>
            <constraint firstItem="mqx-Yb-8Le" firstAttribute="top" secondItem="
0il-CG-C0J" secondAttribute="bottom" constant="8" id="GVY-cS-H6m"/>
            <constraint firstAttribute="trailing" secondItem="0il-CG-C0J"
secondAttribute="trailing" constant="8" id="Np8-AB-5jS"/>
            <constraint firstItem="mqx-Yb-8Le" firstAttribute="leading" secondItem="
6lI-3h-wdU" secondAttribute="leading" constant="8" id="Qdk-gM-NYx"/>
            <constraint firstAttribute="bottom" secondItem="mqx-Yb-8Le"
secondAttribute="bottom" constant="9" id="SCI-pm-nUu"/>
            <constraint firstItem="mqx-Yb-8Le" firstAttribute="height" secondItem="
0il-CG-C0J" secondAttribute="height" id="VYE-2x-g5D"/>
            <constraint firstItem="0il-CG-C0J" firstAttribute="leading" secondItem="
6lI-3h-wdU" secondAttribute="leading" constant="8" id="cyN-04-IRQ"/>
            <constraint firstItem="0il-CG-C0J" firstAttribute="top" secondItem="
6lI-3h-wdU" secondAttribute="top" id="qx5-GP-Buc"/>
        </constraints>
    </view>
    </subviews>
    <color key="backgroundColor" white="1" alpha="1" colorSpace="custom"
customColorSpace="calibratedWhite"/>
    <constraints>
        <constraint firstItem="wfy-db-euE" firstAttribute="top" secondItem="6lI-3h-wdU"
secondAttribute="bottom" id="0fO-md-I5x"/>
        <constraint firstItem="QAv-yD-BWh" firstAttribute="top" secondItem="xu3-cC-StE"
secondAttribute="bottom" id="0t5-QZ-FQC"/>
        <constraint firstAttribute="trailingMargin" secondItem="SOj-1k-9zM"
secondAttribute="trailing" constant="-20" id="B7H-6Z-urT"/>
        <constraint firstItem="SOj-1k-9zM" firstAttribute="leading" secondItem="
8bC-Xf-vdC" secondAttribute="leadingMargin" constant="-20" id="GQP-BF-33e"/>
        <constraint firstItem="SOj-1k-9zM" firstAttribute="top" secondItem="y3c-jy-aDJ"

```

```

secondAttribute="bottom" constant="8" id="GrW-nM-U4n"/>
    <constraint firstAttribute="trailingMargin" secondItem="6lI-3h-wdU"
secondAttribute="trailing" constant="-20" id="Mdf-DG-ZRg"/>
    <constraint firstAttribute="trailingMargin" secondItem="xu3-cC-StE"
secondAttribute="trailing" constant="-20" id="Qel-5F-7MT"/>
    <constraint firstItem="xu3-cC-StE" firstAttribute="leading" secondItem="
8bC-Xf-vdC" secondAttribute="leadingMargin" constant="-20" id="Wfv-fc-poV"/>
    <constraint firstAttribute="trailingMargin" secondItem="QAv-yD-BWh"
secondAttribute="trailing" constant="-20" id="aME-Me-cgY"/>
    <constraint firstItem="6lI-3h-wdU" firstAttribute="top" secondItem="QAv-yD-BWh"
secondAttribute="bottom" constant="8" id="abJ-5t-rQF"/>
    <constraint firstItem="6lI-3h-wdU" firstAttribute="leading" secondItem="
8bC-Xf-vdC" secondAttribute="leadingMargin" constant="-20" id="u95-fe-xMw"/>
    <constraint firstItem="xu3-cC-StE" firstAttribute="top" secondItem="SOj-1k-9zM"
secondAttribute="bottom" id="vSf-9Y-nXY"/>
    <constraint firstItem="QAv-yD-BWh" firstAttribute="leading" secondItem="
8bC-Xf-vdC" secondAttribute="leadingMargin" constant="-20" id="yDw-Ws-Ktq"/>
    </constraints>
</view>
<connections>
    <outlet property="connectionStatus" destination="xu3-cC-StE" id="M3h-ab-ZmD"/>
    <outlet property="lcdTextView" destination="0il-CG-C0J" id="HfP-ij-gt7"/>
    <outlet property="logTextView" destination="mqx-Yb-8Le" id="fLO-Kx-aEj"/>
</connections>
</viewController>
<placeholder placeholderIdentifier="IBFirstResponder" id="dkx-z0-nzr" sceneMemberID="
firstResponder"/>
</objects>
<point key="canvasLocation" x="708" y="437"/>
</scene>
</scenes>
</document>

```

### 7.8.5 Step 5: Finalize the View Controller File

In the view controller file, perform the following:

- Set delegate and initialize [IDT\\_UniPayI\\_V](#) singleton object in the viewDidLoad method. Reference: [Allocate/Initialize IDT\\_UniPayI\\_V Object](#)

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    IDT_UniPayI_V.sharedController().delegate = self
}

```

- Implement protocol delegate [IDT\\_UniPayI\\_VDelegate::deviceDisconnected\(\)](#) and [IDT\\_UniPayI\\_VDelegate::deviceConnected\(\)](#) to monitor connect/disconnect events and modify our connection label upon change. Reference: [Implement Optional Delegate Protocols](#)

```

func setConnectionStatus(status: String, backgroundColor: UIColor) {
    connectionStatus.text = status;
    connectionStatus.backgroundColor = backgroundColor
}

func appendMessageToLCD(message: String) {
    lcdTextView.text = "\n===== \n(message)\n(lcdTextView.text)"
    lcdTextView.scrollToVisible(NSRange(location: 0, length: 0))
}

func deviceConnected() {
    setConnectionStatus("Connected", backgroundColor: UIColor.greenColor())
    appendMessageToLCD("UniPay 1.5 Connected\nFramework Version: \(IDT_Device.SDK_version())")
}

func deviceDisconnected() {
    setConnectionStatus("Disconnected", backgroundColor: UIColor.redColor())
}

```

- Implement protocol delegate `swipeMSRData()` to receive card swipe data. Reference: [Implement Optional Delegate Protocols](#)

```
func swipeMSRData(cardData: IDTMSRData!) {
    NSLog("--MSR event received, type: \(cardData.event), data: \(cardData.encTrack1)")
    switch cardData.event {
    case EVENT_MSR_CARD_DATA:
        switch cardData.captureEncodeType {
        case CAPTURE_ENCODE_TYPE_ISOABA:
            appendMessageToLCD("Encode Type: ISO/ABA")
        case CAPTURE_ENCODE_TYPE_AAMVA:
            appendMessageToLCD("Encode Type: AA/MVA")
        case CAPTURE_ENCODE_TYPE_Other:
            appendMessageToLCD("Encode Type: Other")
        case CAPTURE_ENCODE_TYPE_Raw:
            appendMessageToLCD("Encode Type: Raw")
        case CAPTURE_ENCODE_TYPE_JIS_I:
            appendMessageToLCD("Encode Type: CAPTURE_ENCODE_TYPE_JIS_I")
        case CAPTURE_ENCODE_TYPE_JIS_II:
            appendMessageToLCD("Encode Type: CAPTURE_ENCODE_TYPE_JIS_II")
        default:
            appendMessageToLCD("Encode Type: UNKWOWN")
        }

        switch cardData.captureEncryptType {
        case CAPTURE_ENCRYPT_TYPE_AES:
            appendMessageToLCD("Encrypt Type: AES")
        case CAPTURE_ENCRYPT_TYPE_TDES:
            appendMessageToLCD("Encrypt Type: TDES")
        case CAPTURE_ENCRYPT_TYPE_NO_ENCRYPTION:
            appendMessageToLCD("Encrypt Type: NONE")
        default:
            appendMessageToLCD("Encrypt Type: UNKNOWN")
        }

        appendMessageToLCD("Full card data: \(cardData.cardData == nil ? "N/A" : cardData.cardData)")
        appendMessageToLCD("Track 1: \(cardData.track1 == nil ? "N/A" : cardData.track1)")
        appendMessageToLCD("Track 2: \(cardData.track2 == nil ? "N/A" : cardData.track2)")
        appendMessageToLCD("Track 3: \(cardData.track3 == nil ? "N/A" : cardData.track3)")
        appendMessageToLCD("Length Track 1: \(cardData.track1Length)")
        appendMessageToLCD("Length Track 2: \(cardData.track2Length)")
        appendMessageToLCD("Length Track 3: \(cardData.track3Length)")
        appendMessageToLCD("Encoded Track 1: \(cardData.encTrack1 == nil ? "N/A" : cardData.encTrack1.description)")
        appendMessageToLCD("Encoded Track 2: \(cardData.encTrack2 == nil ? "N/A" : cardData.encTrack2.description)")
        appendMessageToLCD("Encoded Track 3: \(cardData.encTrack3 == nil ? "N/A" : cardData.encTrack3.description)")
        appendMessageToLCD("Hash Track 1: \(cardData.hashTrack1 == nil ? "N/A" : cardData.hashTrack1.description)")
        appendMessageToLCD("Hash Track 2: \(cardData.hashTrack2 == nil ? "N/A" : cardData.hashTrack2.description)")
        appendMessageToLCD("Hash Track 3: \(cardData.hashTrack3 == nil ? "N/A" : cardData.hashTrack3.description)")
        appendMessageToLCD("KSN: \(cardData.KSN == nil ? "N/A" : cardData.KSN.description)")
        appendMessageToLCD("\nSessionID: \(cardData.sessionID == nil ? "N/A" : cardData.sessionID.description)")
        appendMessageToLCD("\nReader Serial Number: \(cardData.RSN == nil ? "N/A" : cardData.RSN)")
        appendMessageToLCD("\nRead Status: \(cardData.readStatus)")

        if cardData.unencryptedTags != nil {
            appendMessageToLCD("Unencrypted Tags: \(cardData.unencryptedTags.description)")
        }

        if cardData.encryptedTags != nil {
            appendMessageToLCD("Encrypted Tags: \(cardData.encryptedTags.description)")
        }

        if cardData.maskedTags != nil {
            appendMessageToLCD("Masked Tags: \(cardData.maskedTags.description)")
        }

        NSLog("Track 1: \(cardData.track1 == nil ? "N/A" : cardData.track1)")
        NSLog("Track 2: \(cardData.track2 == nil ? "N/A" : cardData.track2)")
        NSLog("Track 3: \(cardData.track3 == nil ? "N/A" : cardData.track3)")
        NSLog("Encoded Track 1: \(cardData.encTrack1 == nil ? "N/A" : cardData.encTrack1.description)")
        NSLog("Encoded Track 2: \(cardData.encTrack2 == nil ? "N/A" : cardData.encTrack2.description)")
        NSLog("Encoded Track 3: \(cardData.encTrack3 == nil ? "N/A" : cardData.encTrack3.description)")
        NSLog("Hash Track 1: \(cardData.hashTrack1 == nil ? "N/A" : cardData.hashTrack1.description)")
        NSLog("Hash Track 2: \(cardData.hashTrack2 == nil ? "N/A" : cardData.hashTrack2.description)")
        NSLog("Hash Track 3: \(cardData.hashTrack3 == nil ? "N/A" : cardData.hashTrack3.description)")
    }
}
```

```

        NSLog("SessionID: \(cardData.sessionID == nil ? "N/A" : cardData.sessionID.description)")
        NSLog("nReader Serial Number: \(cardData.RSN == nil ? "N/A" : cardData.RSN)")
        NSLog("Read Status: \(cardData.readStatus)")
        NSLog("KSN: \(cardData.KSN == nil ? "N/A" : cardData.KSN.description)")

        case EVENT_MSR_CANCEL_KEY:
            appendMessageToLCD("(Event) MSR Cancel Key received: \(cardData.encTrack1)")

        case EVENT_MSR_BACKSPACE_KEY:
            appendMessageToLCD("(Event) MSR Backspace Key received: \(cardData.encTrack1)")

        case EVENT_MSR_ENTER_KEY:
            appendMessageToLCD("(Event) MSR Enter Key received: \(cardData.encTrack1)")

        case EVENT_MSR_UNKNOWN:
            appendMessageToLCD("(Event) MSR unknown event, data: \(cardData.encTrack1)")

        default:
            break
    }
}

```

- Implement protocol delegate `plugStatusChange()` to automatically attempt connection. Reference: [Implement Optional Delegate Protocols](#)

```

func plugStatusChange(deviceInserted: Bool) {
    if deviceInserted {
        appendMessageToLCD("Device attached. Attempting to connect...")
        IDT_UniPayI_V.sharedController().device_connectToAudioReader()
    } else {
        appendMessageToLCD("Device removed.")
    }
}

```

- Implement protocol delegate `emvTransactionData()` to report EMV transaction results. Reference: [Implement Optional Delegate Protocols](#)

```

func emvTransactionData(emvData: IDTEMVData!, errorCode error: Int32) {
    appendMessageToLCD("EMV transaction data response: \(
        IDT_UniPayI_V.sharedController().device_getResponseCodeString(error))\n")

    if emvData == nil {
        appendMessageToLCD("EMV TRANSACTION ERROR. Refer to EMV_RESULT_CODE_V2_response = \(error)")
        return;
    }

    if emvData.resultCodeV2 != EMV_RESULT_CODE_V2_NO_RESPONSE {
        appendMessageToLCD("EMV_RESULT_CODE_V2_RESPONSE: \(emvData.resultCodeV2.rawValue)")
    }

    if emvData.resultCodeV2 == EMV_RESULT_CODE_V2_GO_ONLINE {
        appendMessageToLCD("ONLINE REQUEST")
    }

    if emvData.resultCodeV2 == EMV_RESULT_CODE_V2_START_TRANS_SUCCESS {
        appendMessageToLCD("Start success: authentication required")
    }

    if emvData.resultCodeV2 == EMV_RESULT_CODE_V2_APPROVED || emvData.resultCodeV2 ==
        EMV_RESULT_CODE_V2_APPROVED_OFFLINE {
        appendMessageToLCD("APPROVED");
    }

    if emvData.resultCodeV2 == EMV_RESULT_CODE_V2_MSR_SUCCESS {
        appendMessageToLCD("MSR Data Captured")
    }

    if emvData.cardType == 0 {
        appendMessageToLCD("CONTACT")
    }
}

```

```

    if emvData.cardType == 1 {
        appendMessageToLCD("CONTACTLESS")
    }

    if emvData.unencryptedTags != nil {
        appendMessageToLCD("Unencrypted Tags: \(emvData.unencryptedTags.description)")
    }

    if emvData.encryptedTags != nil {
        appendMessageToLCD("Encrypted Tags: \(emvData.encryptedTags.description)")
    }

    if emvData.maskedTags != nil {
        appendMessageToLCD("Masked Tags: \(emvData.maskedTags.description)")
    }

    if emvData.hasAdvise {
        appendMessageToLCD("Response has advise request")
    }

    if emvData.hasReversal {
        appendMessageToLCD("Response has reversal request")
    }
}

```

- Implement protocol delegate `lcdDisplay:()` to receive LCD messages, and automatically select 1st menu item/language when presented with choices. Normal operation would require a choice be made by card holder. Reference: [Implement Optional Delegate Protocols](#)

```

func lcdDisplay(mode: Int32, lines: [AnyObject]!) {
    var str = ""

    if lines != nil {
        for s in lines {
            str += s as! String
            str += "\n"
        }
    }

    switch mode {
        case 0x10:
            lcdTextView.text = ""
        case 0x03:
            lcdTextView.text = str
        case 0x01, 0x02, 0x08:
            IDT_UniPayI_V.sharedController().emv_callbackResponseLCD(mode,
                selection: 1)
        default:
            break
    }
}

```

- Implement the button press methods

```

func displayReturnError(operation: String, rt: RETURN_CODE) {
    let message = "\(operation) ERROR: ID-\(rt.rawValue), Message: \(
        IDT_UniPayI_V.sharedController().device_getResponseCodeString(Int32(rt.rawValue))) "
    appendMessageToLCD(message)
}

@IBAction func getFirmware(sender: UIButton) {
    var result: NSString?
    let rt = IDT_UniPayI_V.sharedController().device_getFirmwareVersion(&
        result)

    if RETURN_CODE_DO_SUCCESS == rt {
        appendMessageToLCD("Get firmware: \(result!)")
    } else {
        displayReturnError("Get firmware", rt: rt)
    }
}

```

```

@IBAction func startMSR(sender: UIButton) {
    let rt = IDT_UniPayI_V.sharedController().
        msr_startMSRSwipe()

    if RETURN_CODE_DO_SUCCESS == rt {
        appendMessageToLCD("Started MSR")
    } else {
        displayReturnError("Start MSR", rt: rt)
    }
}

@IBAction func startICCEMV(sender: UIButton) {
    let rt = IDT_UniPayI_V.sharedController().emv_startTransaction(1.00,
        amtOther: 0, type: 0, timeout: 60, tags: nil, forceOnline: false, fallback: true)

    if RETURN_CODE_DO_SUCCESS == rt {
        appendMessageToLCD("Start transaction command accepted")
    } else {
        displayReturnError("Start ICC EMV", rt: rt)
    }
}

@IBAction func completeICCEMV(sender: UIButton) {
    let rt = IDT_UniPayI_V.sharedController().emv_completeOnlineEMVTransaction
        (true, hostResponseTags: IDTUtility.hexToData("8A023030"))

    if RETURN_CODE_DO_SUCCESS == rt {
        appendMessageToLCD("Complete transaction command accepted")
    } else {
        displayReturnError("Complete ICC EMV", rt: rt)
    }
}

@IBAction func cancelTransaction(sender: UIButton) {
    let rt = IDT_UniPayI_V.sharedController().
        msr_cancelMSRSwipe()

    if RETURN_CODE_DO_SUCCESS == rt {
        appendMessageToLCD("Cancelled MSR")
    } else {
        displayReturnError("Cancel transaction", rt: rt)
    }
}

@IBAction func clearOutputs(sender: UIBarButtonItem) {
    lcdTextView.text = "";
    logTextView.text = "";
}

```



## Chapter 8

# UniPay 1.5 Error Code Reference

0000	OK
0001	Incorrect Header Tag
0002	Unknown Command
0003	Unknown Sub-Command
0004	CRC Error in Frame
0005	Incorrect Parameter
0006	Parameter Not Supported
0007	Mal-formatted Data
0008	Timeout
000A	Failed / NACK
000B	Command not Allowed
000C	Sub-Command not Allowed
000D	Buffer Overflow (Data Length too large for reader buffer)
000E	User Interface Event
0011	Communication type not supported, VT-1, burst, etc.
0012	Secure interface is not functional or is in an intermediate state.
0013	Data field is not mod 8
0014	Pad 0x80 not found where expected
0015	Specified key type is invalid
0016	Could not retrieve key from the SAM (InitSecureComm)
0017	Hash code problem
0018	Could not store the key into the SAM (InstallKey)
0019	Frame is too large
001A	Unit powered up in authentication state but POS must resend the InitSecureComm command
001B	The EEPROM may not be initialized because SecCommInterface does not make sense
001C	Problem encoding APDU
0020	Unsupported Index (ILM) SAM Transceiver error - problem communicating with the SAM (Key Mgr)
0021	Unexpected Sequence Counter in multiple frames for single bitmap (ILM) Length error in data returned from the SAM (Key Mgr)
0022	Improper bit map (ILM)
0023	Request Online Authorization
0024	ViVOCard3 raw data read successful
0025	Message index not available (ILM) ViVOcomm activate transaction card type (ViVOcomm)
0026	Version Information Mismatch (ILM)
0027	Not sending commands in correct index message index (ILM)
0028	Time out or next expected message not received (ILM)
0029	ILM languages not available for viewing (ILM)
002A	Other language not supported (ILM)
0050	Auto-Switch OK
0051	Auto-Switch failed
0060	Data not exist
0061	Data Full
0062	Write Flash Error
0063	Ok and Have Next Command
0090	Account DUKPT Key not exist
0091	Account DUKPT Key KSN exhausted
EE00	OK
EE01	Incorrect Header Tag
EE02	Unknown Command
EE03	Unknown Sub-Command
EE04	CRC Error in Frame
EE05	Incorrect Parameter
EE06	Parameter Not Supported
EE07	Mal-formatted Data
EE08	Timeout
EE0A	Failed / NACK
EE0B	Command not Allowed
EE0C	Sub-Command not Allowed
EE0D	Buffer Overflow (Data Length too large for reader buffer)

EE0E	User Interface Event
EE11	Communication type not supported, VT-1, burst, etc.
EE12	Secure interface is not functional or is in an intermediate state.
EE13	Data field is not mod 8
EE14	Pad 0x80 not found where expected
EE15	Specified key type is invalid
EE16	Could not retrieve key from the SAM (InitSecureComm)
EE17	Hash code problem
EE18	Could not store the key into the SAM (InstallKey)
EE19	Frame is too large
EE1A	Unit powered up in authentication state but POS must resend the InitSecureComm command
EE1B	The EEPROM may not be initialized because SecCommInterface does not make sense
EE1C	Problem encoding APDU
EE20	Unsupported Index (ILM) SAM Transceiver error - problem communicating with the SAM (Key Mgr)
EE21	Unexpected Sequence Counter in multiple frames for single bitmap (ILM) Length error in data returned from the SAM (Key Mgr)
EE22	Improper bit map (ILM)
EE23	Request Online Authorization
EE24	ViVOCard3 raw data read successful
EE25	Message index not available (ILM) ViVOcomm activate transaction card type (ViVOcomm)
EE26	Version Information Mismatch (ILM)
EE27	Not sending commands in correct index message index (ILM)
EE28	Time out or next expected message not received (ILM)
EE29	ILM languages not available for viewing (ILM)
EE2A	Other language not supported (ILM)
EE50	Auto-Switch OK
EE51	Auto-Switch failed
EE60	Data not exist
EE61	Data Full
EE62	Write Flash Error
EE63	Ok and Have Next Command
EE90	Account DUKPT Key not exist
EE91	Account DUKPT Key KSN exhausted

## Chapter 9

# Enumeration Reference

### 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{
    IDT_DEVICE_BTPAY_IOS = 0,
    IDT_DEVICE_BTPAY_OSX_BT,
    IDT_DEVICE_BTPAY_OSX_USB,
    IDT_DEVICE_UNIPAY_IOS,
    IDT_DEVICE_UNIPAY_OSX_USB,
    IDT_DEVICE_UniPayIII_IOS,
    IDT_DEVICE_UniPayIII_OSX_USB,
    IDT_DEVICE_IMAG_IOS,
    IDT_DEVICE_VENDI_MOBILE,
    IDT_DEVICE_UniPayI_V_IOS,
    IDT_DEVICE_UniPayI_V_OSX_USB
}IDT_DEVICE_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_UniPayIII_EMV_NO_ICC_MSR_DATA,
    EVENT_UniPayIII_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_TIMEOUT,
    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_V2_APPROVED_OFFLINE = 0x0000,
    EMV_RESULT_CODE_V2_DECLINED_OFFLINE = 0x0001,
    EMV_RESULT_CODE_V2_APPROVED = 0x0002,
    EMV_RESULT_CODE_V2_DECLINED = 0x0003,
    EMV_RESULT_CODE_V2_GO_ONLINE = 0x0004,
    EMV_RESULT_CODE_V2_CALL_YOUR_BANK = 0x0005,
    EMV_RESULT_CODE_V2_NOT_ACCEPTED = 0x0006,
    EMV_RESULT_CODE_V2_USE_MAGSTRIPE = 0x0007,
    EMV_RESULT_CODE_V2_TIME_OUT = 0x0008,
    EMV_RESULT_CODE_V2_START_TRANS_SUCCESS = 0x0010,
    EMV_RESULT_CODE_V2_MSR_SUCCESS = 0x0011,
    EMV_RESULT_CODE_V2_FILE_ARG_INVALID = 0x1001,
    EMV_RESULT_CODE_V2_FILE_OPEN_FAILED = 0x1002,
    EMV_RESULT_CODE_V2_FILE_OPERATION_FAILED = 0x1003,
    EMV_RESULT_CODE_V2_MEMORY_NOT_ENOUGH = 0x2001,
    EMV_RESULT_CODE_V2_SMARTCARD_FAIL = 0x3001,
    EMV_RESULT_CODE_V2_SMARTCARD_INIT_FAILED = 0x3003,
    EMV_RESULT_CODE_V2_FALLBACK_SITUATION = 0x3004,
    EMV_RESULT_CODE_V2_SMARTCARD_ABSENT = 0x3005,
    EMV_RESULT_CODE_V2_SMARTCARD_TIMEOUT = 0x3006,
    EMV_RESULT_CODE_V2_MSR_CARD_ERROR = 0x3007,
    EMV_RESULT_CODE_V2_PARSING_TAGS_FAILED = 0x5001,
    EMV_RESULT_CODE_V2_CARD_DATA_ELEMENT_DUPLICATE = 0x5002,
    EMV_RESULT_CODE_V2_DATA_FORMAT_INCORRECT = 0x5003,
    EMV_RESULT_CODE_V2_APP_NO_TERM = 0x5004,
    EMV_RESULT_CODE_V2_APP_NO_MATCHING = 0x5005,
    EMV_RESULT_CODE_V2_AMANDATORY_OBJECT_MISSING = 0x5006,
    EMV_RESULT_CODE_V2_APP_SELECTION_RETRY = 0x5007,
    EMV_RESULT_CODE_V2_AMOUNT_ERROR_GET = 0x5008,
    EMV_RESULT_CODE_V2_CARD_REJECTED = 0x5009,
    EMV_RESULT_CODE_V2_AIP_NOT_RECEIVED = 0x5010,
    EMV_RESULT_CODE_V2_AFL_NOT_RECEIVEDE = 0x5011,
    EMV_RESULT_CODE_V2_AFL_LEN_OUT_OF_RANGE = 0x5012,
    EMV_RESULT_CODE_V2_SFI_OUT_OF_RANGE = 0x5013,
    EMV_RESULT_CODE_V2_AFL_INCORRECT = 0x5014,
    EMV_RESULT_CODE_V2_EXP_DATE_INCORRECT = 0x5015,
    EMV_RESULT_CODE_V2_EFF_DATE_INCORRECT = 0x5016,
    EMV_RESULT_CODE_V2_ISS_COD_TBL_OUT_OF_RANGE = 0x5017,
    EMV_RESULT_CODE_V2_CRYPTOGAM_TYPE_INCORRECT = 0x5018,
    EMV_RESULT_CODE_V2_PSE_BY_CARD_NOT_SUPPORTED = 0x5019,
    EMV_RESULT_CODE_V2_USER_LANGUAGE_SELECTED = 0x5020,
    EMV_RESULT_CODE_V2_SERVICE_NOT_ALLOWED = 0x5021,
    EMV_RESULT_CODE_V2_NO_TAG_FOUND = 0x5022,
    EMV_RESULT_CODE_V2_CARD_BLOCKED = 0x5023,
    EMV_RESULT_CODE_V2_LEN_INCORRECT = 0x5024,
    EMV_RESULT_CODE_V2_CARD_COM_ERROR = 0x5025,
    EMV_RESULT_CODE_V2_TSC_NOT_INCREASED = 0x5026,
    EMV_RESULT_CODE_V2_HASH_INCORRECT = 0x5027,
    EMV_RESULT_CODE_V2_ARC_NOT_PRESENCED = 0x5028,
    EMV_RESULT_CODE_V2_ARC_INVALID = 0x5029,
    EMV_RESULT_CODE_V2_COMM_NO_ONLINE = 0x5030,
    EMV_RESULT_CODE_V2_TRAN_TYPE_INCORRECT = 0x5031,
    EMV_RESULT_CODE_V2_APP_NO_SUPPORT = 0x5032,
    EMV_RESULT_CODE_V2_APP_NOT_SELECT = 0x5033,

```

```
EMV_RESULT_CODE_V2_LANG_NOT_SELECT = 0X5034,  
EMV_RESULT_CODE_V2_TERM_DATA_NOT_PRESENCE = 0X5035,  
EMV_RESULT_CODE_V2_CVM_TYPE_UNKNOWN = 0X6001,  
EMV_RESULT_CODE_V2_CVM_AIP_NOT_SUPPORTED = 0X6002,  
EMV_RESULT_CODE_V2_CVM_TAG_8E_MISSING = 0X6003,  
EMV_RESULT_CODE_V2_CVM_TAG_8E_FORMAT_ERROR = 0X6004,  
EMV_RESULT_CODE_V2_CVM_CODE_IS_NOT_SUPPORTED = 0X6005,  
EMV_RESULT_CODE_V2_CVM_COND_CODE_IS_NOT_SUPPORTED = 0X6006,  
EMV_RESULT_CODE_V2_CVM_NO_MORE = 0X6007,  
EMV_RESULT_CODE_V2_PIN_BYPASSED_BEFORE = 0X6008  
} EMV_RESULT_CODE_V2_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 10

# 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 cumulative 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
DFEE01	ApplePay VAS Protocol
DFEE02	ApplePay VAS Failure Report
DFEE03	ViVotech Proprietary Suite
DFEE04	TAC Online
DFEE05	Threshold Value for Biased Random Selection.
DFEE06	Target Percentage for Random Transaction Selection
DFEE07	Maximum Target Percentage for Random Transaction Selection
DFEE08	RID (in AR)
DFEE09	Last 4 digits of Primary Account Number (PAN)
DFEE0A	Contactless Capabilities (Visa Transit)
DFEE0B	Issuer Script Results
DFEE0C	Issuer Script Results
DFEE0D	Force Transaction Online
DFEE0E	Default DDOL
DFEE0F	Enable Revocation List Processing
DFEE10	Terminal Languages Supported
DFEE11	Enable Transaction Logging
DFEE12	KSN of Account DUKPT Key
DFEE12	KSN of Account DUKPT Key
DFEE13	TAC Default
DFEE14	TAC Denial
DFEE15	Application Selection Indicator
DFEE16	DUKPT Key or MKSK Select for Online PIN Encrypted
DFEE17	ICC Terminal Entry Mode
DFEE18	MSR Terminal Entry Mode
DFEE19	Online DOL
DFEE1A	Output data element
DFEE1B	Authorization Request data elements
DFEE1C	LCD Font Size
DFEE1D	LCD delay Time
DFEE1E	Terminal Configuration
DFEE1F	Issuer Script Limit
DFEE20	ICC power on waiting time
DFEE21	ICC L1 data transaction waiting time
DFEE22	Driver (Menu, Get PIN, Get MSR) Timeout

Tag	Description
DFEE23	MSR all track data
DFEE24	Force Acceptance
DFEE25	ICC Response Code
DFEE26	Encryption Status Information
DFEE27	MSR Control
DFEE28	Terminal Capabilities - No CVM Required
DFEE29	Terminal Capabilities - CVM Required
DFEE2A	Threshold Value for Biased Random Selection (Interac)
DFEE2B	Maximum Target Percentage for Biased Random Selection (Interac)
DFEE2C	Target Percentage for Random Selection (Interac)
DFEE2D	Group Number / Fallback Group
DFEE2E	Max AID Length
DFEE2F	AID Disabled
DFEE30	Track Data Source
DFEE31	DD Card Track 1
DFEE32	DD Card Track 2
DFEE33	Interac Receipt Required
DFEE34	Terminal Contactless Transaction Limit
DFEE35	Visa Reader Risk Flags
DFEE36	CVM Required Limit
DFEE37	LED Color
DFEE38	Language Option for LCD
DFEE39	Force MagStripe
DFEE3A	TAC - Online
DFEE3B	TAC - Default
DFEE3C	TAC - Denial
DFEE3D	Reader Contactless Floor Limit Data
DFEE3E	Enable Exception List Processing
DFEE3F	Default TDOL
DFEE40	Message to be displayed by EMV Kernel on "PIN Try Limit Exceeded" condition
DFEE41	Message to be displayed by EMV Kernel on "Last PIN Try" condition
DFEE42	Message to be displayed by EMV Kernel on "Please Try Again" condition
DFEE43	Message to be displayed by EMV Kernel on "Call Your Bank" condition
DFEE44	Application Capability
DFEE45	GMEDS Secret Keys
DFEE46	GMAD MIDs
DFEE47	ISIS Read Cmd Data
DFEE48	ISIS Write Data
DFEE49	ISIS Transaction Data
DFEE4A	Registered Application Provider Identifier (RID)
DFEE4B	Partial Selection Allowed
DFEE4C	Application Flow
DFEE4D	Selection Features - GR 1.2.10
DFEE4E	Polling Options
DFEE4F	Interface Support
DFEE50	Special Flow
DFEE51	Amex Terminal Capability (used for Amex only)
DFEE52	Transaction CVM
DFEE53	Exclude from Processing
DFEE54	Kernel ID Transaction Type Group List

Tag	Description
DFEE55	RID
DFEE56	Activate Trans for DESFireViVOCComm Flows
DFEE57	Reader Primary Language
DFEE58	Reader Secondary Language
DFEE59	Default Kernel ID
DFEE5A	TLV Exclusion List
DFEE5B	Terminal Entry Capability
DFEE5C	RF Deactivate Period
DFEE5D	D-PAS Issuer Script Response status
DFEE5E	Transaction Timing Information
DFEE5F	Encrypted PAN for remote PIN Pad
DFEE60	Product ID
DFEE61	Processor ID
DFEE62	Main Firmware Build ID
DFEE63	CB Enhanced DDA Indicator (same block as DF03)
DFEE64	CB Wave 2 CVM Requirements (same block as DF04)
DFEE65	Build ID Num (Cxx)
DFEE66	SVN Number
DFEE67	Specific Features Switch
DFEE68	Enable/Disable STOP command processing
DFEE69	ConfigureProprietaryTags
DFEE6A	Enable/Disable Comm Error Recovery
DFEE6B	Terminal IFD
DFEE6C	Cubic FTP Phase 2 Mode Options
DFEE6D	Cubic Mode 3 Match AID
DFEE6E	Torn Transaction Log Clean Interval (minutes)
DFEE6F	Cubic Timestamp Data
DFEE70	Loyalty Program ID
DFEE71	Value Added Tax 1
DFEE72	Value Added Tax 2
DFEE73	Merchant Category Code
DFEE74	Discover Optional Features
DFEE75	Communications Error Message Delay
DFEE76	TVR from GenAC
DFEE77	ViVOpay MSR Custom Data Output Tag
DFEE78	MC Timing Performance Enable
DFEE79	Card Disable Mask
DFEE7A	Card Disable Interval
DFEE7B	Serial Port (UART) Inter-character Timeout Period
DFEE7C	Auto Switch Feature
DFEE7D	Track Formatting Feature
DFEE7E	Burst Mode
DFEE7F	Improved Collision Detection & Media Removal Feature
DFEF01	2nd usage: Remaining Candidates
DFEF02	2nd usage: Single application flow in all candidates flag
DFEF03	GMEDs Data
DFEF04	MSR Encryption Option
DFEF05	CVMRequiredLimit_JCBScheme
DFEF06	CB Display Offline Funds Indicator (same block as DF05)
DFEF07	CB Terminal Type (same block as 9F35)

Tag	Description
DFEF08	Generic Name String
DFEF09	Serial Finite State Machine Version
DFEF0A	Generic Numeric
DFEF0B	Generic Specification String
DFEF0C	System Information Suite
DFEF0D	Generic Implementation String
DFEF0E	Serial Protocol Version
DFEF0F	Serial Protocol Suite
DFEF10	L1 Paypass Version
DFEF11	L1 LCR Version
DFEF12	VIUDS Scheme IDs Supported
DFEF13	VIUDS Scheme ID Selection Criteria
DFEF14	Transaction Finite State Machine Version
DFEF15	L2 Card App Version
DFEF16	M/Chip3 Intermediate Message Marker
DFEF17	Track 1 Data
DFEF18	Track 2 Data
DFEF19	Unpredictable Number Range
DFEF1A	Configuration of encrypting and masking data
DFEF1B	L2 Card App Suite
DFEF1C	User Experience Version
DFEF1D	User Experience Suite
DFEF1E	Encrypted Sensitive Tags
DFEF1F	Auto Authenticate
DFEF20	MAC option in reponse data
DFEF21	BIN
DFEF22	AID
DFEF23	HMAC
DFEF24	HMAC KSN
DFEF25	Next available TLV
FFEE01	ViVopay TLV Group Tag
FFEE02	ViVopay Pre-PPSE Special Flow Group Tag
FFEE03	ViVopay Post-PPSE Special Flow Group Tag
FFEE04	M/Chip3 Intermediate Message Data
FFEE05	ViVopay MChip Group Status
FFEE06	ApplePay VAS Container
FFEE07	Encrypted Sensitive Tags
FFEE08	Masked Tags
FFEE09	Cubic Fixed Fare Amounts
FFEE0B	AID Range
FFEE0C	White List
FFEE0D	available
FFEE0E	available
FFEE0F	available
FFEE10	ViVopay MChip Group Tag
FFEE11	ViVopay Discover Group Tag

## Chapter 11

# LCD Foreign Language Mapping Table

ID	Message ID	English	French	Spanish	Chinese
0	MSG_NULL	-	-	-	-
1	MSG_AMOUNT	AMOUNT	MONTANT	CANTIDAD	金
2	MSG_AMOUNT_↔ OK	AMOUNT OK?	MONTANT OK	MONTO CORRE↔ CTO?	确定金
3	MSG_APPROVED	APPROVED	APPROUVE	APROVADO	通
4	MSG_CALL_YO↔ UR_BANK	CALL YOUR BANK	APPE VOTRE B↔ ANQE	LLAME A SU BA↔ NCO	系您的行
5	MSG_CANCEL_↔ OR_ENTER	CANCEL OR EN↔ TER	ANNULE OU EN↔ TRER	CANCEL O ENT↔ RAR	取消或确定
6	MSG_CARD_ER↔ ROR	CARD ERROR	ERREUR CARTE	ERROR DE TAR↔ JETA	卡
7	MSG_DECLINED	DECLINED	REFUSE	DECLINADO	卡被拒
8	MSG_ENTER_A↔ MOUNT	ENTER AMOUNT	ENTRER MONT↔ ANT	INGRESE MONTO	入金
9	MSG_ENTER_PIN	ENTER PIN:	ENTRER PIN:	ENTRAR NPI:	入密
10	MSG_INCORRE↔ CT_PIN	INCORRECT PIN	NIP INCORRECT	NPI INCORRECTO	密
11	MSG_ICC_MSR1	SWIPE OR INSE↔ RT	PASSER OU INS↔ ERT	MOVER O INSERT	刷卡或插卡
12	MSG_ICC_MSR2	CARD	CARTE	TARJETA	卡
13	MSG_INSERT_↔ CARD	INSERT CARD	INSERT CARTE	INSERTAR TAR↔ JETA	插卡
14	MSG_USE_CHI↔ P_READER	USE CHIP READ↔ ER UTI	LECTEUR CHIP	USO CHIP LECT↔ OR	使用芯片卡
15	MSG_NOT_ACC↔ EPTED	NOT ACCEPTED	PAS ACCEPTE	DENEGADO	法接受
16	MSG_PIN_OK	GET PIN OK			密正确
17	MSG_PLEASE_↔ WAIT	PLEASE WAIT...	ATTENDRE...	POR FAVOR ES↔ PERE	等候中
18	MSG_PROCESS↔ ING_ERROR	PROCESSING E↔ RROR	ERREUR DE TR↔ AITE	ERROR PROCE↔ SANDO	理
19	MSG_USE_MAG↔ STRIPE	USE MAGSTRIPE	USAGE MAGST↔ RIPE	USO DE MAGST↔ RIPE	使用磁卡
20	MSG_TRY_AGAIN	TRY AGAIN	REESSAYER	VUELV INTENTA↔ RLO	重
21	MSG_ONLINE	GO ONLINE	GO LIGNE	GO LINEA	在



ID	Message ID	English	French	Spanish	Chinese
22	MSG_TRANSACTION_ERROR↔	TRANSACTION ERR	ERREUR DE TRANSACTION	ERROR DE TRANSAC↔	交易
23	MSG_TERMINATE	TERMINATE	RESILIER	TERMINAR	止
24	MSG_ADVICE	ADVICE	CONSEILS	CONSEJOS	建
25	MSG_TIMEOUT	TIME OUT	TIMEOUT	TIEMPO DE ESPERA↔	超
26	MSG_PROCESSING↔	PROCESSING...	PROCESSUS...	PROCESANDO...	理中。。。
27	MSG_PIN_TRY_EX↔	PIN TRY LIMIT EX	PIN TRY DEPASSE	TRY PIN SUPERADA↔	密次多
28	MSG_ISSUER_AUTH_FAIL↔	ISSUER AUTH FAIL	EMETTEUR FAIL	EMISOR FALLA	与卡机构
29	MSG_CONTINUE_PROCESS↔	CONTINUE PROCESS	CONTINUER LA	CONTINUAR PROCES↔	理
30	MSG_GET_PIN_ERROR↔	GET PIN ERROR	GET PIN ERROR	OBTENER PIN ERROR↔	密
31	MSG_GET_PIN_FAIL↔	GET PIN FAIL	GET PIN FAIL	OBTENER PIN FALL↔	取密
32	MSG_NOKEY_GET_PIN↔	NO KEY GET PIN	NO KEY GET PIN	NO CLAVE GET PIN	法入密
33	MSG_CANCELLED↔	CANCELLED	ANNULE	CANCELADO	取消
34	MSG_LAST_PIN_TRY↔	LAST PIN TRY	-	-	最后一次入密

## Chapter 12

# Hierarchical Index

### 12.1 Class Hierarchy

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

AIDEntry . . . . .	76
ApplicationID . . . . .	77
CAKey . . . . .	79
CRLEntry . . . . .	79
ICCReaderStatus . . . . .	80
<IDT_Device_Delegate>	
IDT_UniPayI_V . . . . .	81
MaskAndEncryption . . . . .	109
NSObject	
APDUResponse . . . . .	76
IDT_UniPayI_V . . . . .	81
IDTEMVData . . . . .	104
IDTMSRData . . . . .	106
<NSObject>	
<IDT_UniPayI_V_Delegate> . . . . .	101
PowerOnStructure . . . . .	110
TerminalData . . . . .	110

## Chapter 13

# Class Index

### 13.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AIDEntry	76
APDUResponse	76
ApplicationID	77
CAKey	79
CRLEntry	79
ICCReaderStatus	80
IDT_UniPayI_V	81
<IDT_UniPayI_V_Delegate>	101
IDTEMVData	104
IDTMSRData	106
MaskAndEncryption	109
PowerOnStructure	110
TerminalData	110

## Chapter 14

# Class Documentation

### 14.1 AIDEntry Struct Reference

```
#include <IDTCommon.h>
```

#### Public Attributes

- unsigned char [aid](#) [16]  
*AID value as per payment networks.*
- unsigned char [aidLen](#)  
*AID's length.*

#### 14.1.1 Detailed Description

AID Entry - Used to populate array in IDT\_BTPay::emv\_retrieveAIDList() IDT\_UniPay::emv\_retrieveAIDList().

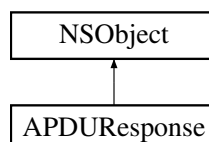
The documentation for this struct was generated from the following file:

- Source\_iOS/IDTCommon.h

### 14.2 APDUResponse Class Reference

```
#import <APDUResponse.h>
```

Inheritance diagram for APDUResponse:



#### Instance Methods

- (void) - [clear](#)

## Class Methods

- ([APDUResponse](#) \*) + [sharedController](#)

## Properties

- unsigned char [SW1](#)  
*Status Word Byte 1.*
- unsigned char [SW2](#)  
*Status Word Byte 2.*
- BOOL [hasKSN](#)  
*KSN data read.*
- BOOL [hasEncryption](#)  
*APDU response is encrypted.*
- int [apduLength](#)  
*Length of valid R-APDU.*
- NSData \* [response](#)  
*APDU Response excluding SW1 and SW2.*
- NSData \* [ksn](#)  
*Key Seral Number.*

### 14.2.1 Detailed Description

Used in IDT\_BTpay::icc\_exchangeAPDU:encrypted:ksn:response:() IDT\_UniPay::icc\_exchangeAPDU:encrypted↵:response:()

### 14.2.2 Method Documentation

#### 14.2.2.1 - (void) clear

clears all [APDUResponse](#) properties

#### 14.2.2.2 + ([APDUResponse](#) \*) sharedController

Singleton instance of [APDUResponse](#)

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

- Source\_iOS/APDUResponse.h

## 14.3 ApplicationID Struct Reference

```
#include <IDTCommon.h>
```

## Public Attributes

- unsigned char [acquirerIdentifier](#) [6]  
*Indicates which acquirer/processor processes the corresponding AID. Tag 9F01.*
- unsigned char [aid](#) [16]  
*AID value as per payment networks. Tag 9F06.*

- unsigned char [aidLen](#)  
*AID's length.*
- unsigned char [applicationSelectionIndicator](#)  
*Standard parameter.*
- unsigned char [applicationVersionNumber](#) [2]  
*EMV application version number. Tag 9F09.*
- unsigned char [XAmount](#) [3]  
*Not used by Agnos Framework.*
- unsigned char [YAmount](#) [3]  
*Not used by Agnos Framework.*
- unsigned char [skipTACIACDefault](#)  
*Indicates whether or not terminal uses default values for risk management.*
- unsigned char [tac](#)  
*Indicates whether or not terminal uses Terminal Action Code. 0x00 or 0x01.*
- unsigned char [floorLimitChecking](#)  
*Indicates whether or not terminal uses Floor Limit Checking. 0x00 or 0x01.*
- unsigned char [randomTransactionSelection](#)  
*Indicates whether or not terminal uses Random Transaction Selection. 0x00 or 0x01.*
- unsigned char [velocityChecking](#)  
*Indicates whether or not terminal uses Velocity Checking. 0x00 or 0x01.*
- unsigned char [tACDenial](#) [5]  
*Terminal Action Code Denial.*
- unsigned char [tACOnline](#) [5]  
*Terminal Action Code Online.*
- unsigned char [tACDefault](#) [5]  
*Terminal Action Code Default.*
- unsigned char [terminalFloorLimit](#) [3]  
*Standard parameter. Tag 9F1B.*
- unsigned char [targetPercentage](#)  
*EMV offline risk management parameter.*
- unsigned char [thresholdValue](#) [3]  
*EMV offline risk management parameter.*
- unsigned char [maxTargetPercentage](#)  
*EMV offline risk management parameter.*
- unsigned char [defaultTDOL](#)  
*Standard parameter.*
- unsigned char [tdolValue](#) [252]  
*Transaction Data Object List value.*
- unsigned char [tdolLen](#)  
*Transaction Data Object List length.*
- unsigned char [defaultDDOL](#)  
*Standard parameter.. Tag.*
- unsigned char [ddolValue](#) [252]  
*Dynamic Data Object List value.*
- unsigned char [ddolLen](#)  
*Dynamic Data Object List length.*
- unsigned char [transactionCurrencyCode](#) [2]  
*AID's currency. Example: For Canada, {0x01,0x24}. Tag 5F2A.*
- unsigned char [transactionCurrencyExponent](#)  
*Transaction Currency Exponent. Example: Amount 4.53\$ is managed as 453. Tag 5F36.*

### 14.3.1 Detailed Description

device AID File - 571 bytes

Used as parameter in IDT\_BTPay::emv\_setApplicationData:()

Used as return value of aidResponse in IDT\_BTPay::emv\_retrieveApplicationData:response:()

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

- Source\_iOS/IDTCommon.h

## 14.4 CAKey Struct Reference

```
#include <IDTCommon.h>
```

### Public Attributes

- unsigned char [hashAlgorithm](#)  
*Hash Algorithm 0x01 = SHA-1.*
- unsigned char [encryptionAlgorithm](#)  
*Encryption Algorithm 0x01 = RSA.*
- unsigned char [rid](#) [5]  
*As per payment networks definition.*
- unsigned char [index](#)  
*As per payment networks definition.*
- unsigned char [exponentLength](#)  
*Length of exponent. 0x01 or 0x03 as per EMV specs.*
- unsigned char [keyLength](#)  
*Length of key. max 248 bytes as per EMV specs.*
- unsigned char [exponent](#) [3]  
*CA Public Key Exponent.*
- unsigned char [key](#) [248]  
*CA Public Key.*

### 14.4.1 Detailed Description

Certificate Authority Public Key

Used as parameter in IDT\_BTPay::emv\_retrieveCAPK:response:(), IDT\_BTPay::emv\_removeCAPK:(), IDT\_BTPay::emv\_setCAPK:(), IDT\_UniPay::emv\_retrieveCAPK:response:(), IDT\_UniPay::emv\_removeCAPK:(), IDT\_UniPay::emv\_setCAPK:()

Used as return value in IDT\_BTPay::emv\_retrieveCAPK:response:() IDT\_UniPay::emv\_retrieveCAPK:response:()

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

- Source\_iOS/IDTCommon.h

## 14.5 CRLEntry Struct Reference

```
#include <IDTCommon.h>
```

## Public Attributes

- unsigned char [rid](#) [5]  
*As per payment networks definition.*
- unsigned char [index](#)  
*As per payment networks definition.*
- unsigned char [serialNumber](#) [3]  
*As per payment networks definition.*

### 14.5.1 Detailed Description

Certificate Revocation List Entry - 9 bytes

Used as parameter in IDT\_BTPay::emv\_retrieveCRLForRID:response(), IDT\_BTPay::emv\_removeCRL:(), IDT\_BTPay::emv\_removeCRLUnit:(), IDT\_BTPay::emv\_setCRL:() IDT\_UniPay::emv\_retrieveCRLForRID:response(), IDT\_UniPay::emv\_removeCRL:(), IDT\_UniPay::emv\_removeCRLUnit:(), IDT\_UniPay::emv\_setCRL:()

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

- Source\_iOS/IDTCommon.h

## 14.6 ICCReaderStatus Struct Reference

```
#include <IDTCommon.h>
```

## Public Attributes

- 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.*

### 14.6.1 Detailed Description

Structure used to return response from IDT\_BTPay::icc\_getICCReaderStatus() and IDT\_UniPay::icc\_getICCReaderStatus()

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

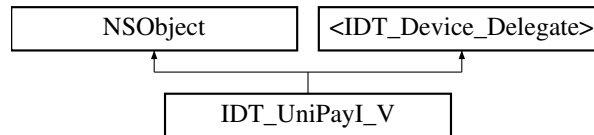
- Source\_iOS/IDTCommon.h



## 14.7 IDT\_UniPayI\_V Class Reference

```
#import <IDT_UniPayI_V.h>
```

Inheritance diagram for IDT\_UniPayI\_V:



### Instance Methods

- (void) - [close](#)
- (void) - [device\\_connectToUSB](#)
- (RETURN\_CODE) - [device\\_cancelConnectToAudioReader](#)
- (RETURN\_CODE) - [device\\_connectToAudioReader](#)
- (RETURN\_CODE) - [device\\_getFirmwareVersion:](#)
- (BOOL) - [device\\_isAudioReaderConnected](#)
- (NSString \*) - [device\\_getResponseCodeString:](#)
- (bool) - [device\\_isConnected:](#)
- (RETURN\_CODE) - [device\\_sendIDGCommand:subCommand:data:response:](#)
- (RETURN\_CODE) - [device\\_setAudioVolume:](#)
- (RETURN\_CODE) - [device\\_setPassThrough:](#)
- (RETURN\_CODE) - [emv\\_authenticateTransaction:](#)
- (RETURN\_CODE) - [device\\_startRKI](#)
- (RETURN\_CODE) - [emv\\_callbackResponseLCD:selection:](#)
- (RETURN\_CODE) - [emv\\_completeOnlineEMVTransaction:hostResponseTags:](#)
- (void) - [emv\\_disableAutoAuthenticateTransaction:](#)
- (RETURN\_CODE) - [emv\\_getEMVL2Version:](#)
- (RETURN\_CODE) - [emv\\_removeApplicationData:](#)
- (RETURN\_CODE) - [emv\\_removeCAPK:index:](#)
- (RETURN\_CODE) - [emv\\_removeCRLList](#)
- (RETURN\_CODE) - [emv\\_removeTerminalData](#)
- (RETURN\_CODE) - [emv\\_retrieveAIDList:](#)
- (RETURN\_CODE) - [emv\\_retrieveApplicationData:response:](#)
- (RETURN\_CODE) - [emv\\_retrieveCAPK:index:response:](#)
- (RETURN\_CODE) - [emv\\_retrieveCAPKFile:index:response:](#)
- (RETURN\_CODE) - [emv\\_retrieveCAPKList:](#)
- (RETURN\_CODE) - [emv\\_retrieveCRLList:](#)
- (RETURN\_CODE) - [emv\\_retrieveTerminalData:](#)
- (RETURN\_CODE) - [emv\\_retrieveTransactionResult:retrievedTags:](#)
- (RETURN\_CODE) - [emv\\_setApplicationData:configData:](#)
- (RETURN\_CODE) - [emv\\_setCAPK:](#)
- (RETURN\_CODE) - [emv\\_setCAPKFile:](#)
- (RETURN\_CODE) - [emv\\_setCRLEntries:](#)
- (RETURN\_CODE) - [emv\\_setTerminalData:](#)
- (RETURN\_CODE) - [emv\\_startTransaction:amtOther:type:timeout:tags:forceOnline:fallback:](#)
- (RETURN\_CODE) - [config\\_getSerialNumber:](#)
- (RETURN\_CODE) - [icc\\_exchangeAPDU:response:](#)
- (RETURN\_CODE) - [icc\\_getICCReaderStatus:](#)
- (RETURN\_CODE) - [icc\\_powerOnICC:](#)
- (RETURN\_CODE) - [icc\\_powerOffICC:](#)
- (RETURN\_CODE) - [msr\\_cancelMSRSwipe](#)
- (RETURN\_CODE) - [msr\\_startMSRSwipe](#)
- (bool) - [isConnected](#)

## Class Methods

- (NSString \*) + [SDK\\_version](#)
- (IDT\_UniPayI\_V \*) + [sharedController](#)

## Properties

- id< [IDT\\_UniPayI\\_V\\_Delegate](#) > [delegate](#)

### 14.7.1 Detailed Description

Class to drive the [IDT\\_UniPayI\\_V](#) device

### 14.7.2 Method Documentation

#### 14.7.2.1 - (void) close

Close Device

#### 14.7.2.2 - (RETURN\_CODE) config\_getSerialNumber: (NSString \*\*) *response*

Polls device for Serial Number

##### Parameters

<i>response</i>	Returns Serial Number
-----------------	-----------------------

##### Returns

RETURN\_CODE: Return codes listed as typedef enum in IDTCommon:RETURN\_CODE. Values can be parsed with IDT\_UniPay::device\_getResponseCodeString:()

#### 14.7.2.3 - (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.

#### 14.7.2.4 - (RETURN\_CODE) device\_connectToAudioReader

Connect To Audio Reader

##### Returns

RETURN\_CODE

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

**14.7.2.5 - (void) device\_connectToUSB**

Connect To Audio Reader

Attempts to recognize and connect to an IDTech MSR device connected via the USB port (OSX only).

**14.7.2.6 - (RETURN\_CODE) device\_getFirmwareVersion: (NSString \*\*) response**

Polls device for Firmware Version

**Parameters**

<i>response</i>	Response returned of Firmware Version
-----------------	---------------------------------------

**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\_UniPay::device\_getResponseCodeString:()

**14.7.2.7 - (NSString \*) device\_getResponseCodeString: (int) errorCode**

Get Response Code String

Interpret a response code and return string description.

**Parameters**

<i>errorCode</i>	Error code, range 0x0000 - 0xFFFF, example 0x0300
------------------	---

**Returns**

Verbose error description

**14.7.2.8 - (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

#### 14.7.2.9 - (bool) device\_isConnected: (IDT\_DEVICE\_Types) device

Is Device Connected

Returns the connection status of the requested device

##### Parameters

<i>device</i>	Check connectivity of device type
---------------	-----------------------------------

```
typedef enum{
    IDT_DEVICE_UniPayI_V_IOS
    IDT_DEVICE_UniPayI_V_OSX_USB
}IDT_DEVICE_Types;
```

#### 14.7.2.10 - (RETURN\_CODE) device\_sendIDGCommand: (unsigned char) command subCommand:(unsigned char) subCommand data:(NSData \*) data response:(NSData \*\*) response

Send NEO IDG Command Send a NEO IDG ViVOtech 2.0 command

##### Parameters

<i>command</i>	One byte command as per NEO IDG Reference Guide
<i>subCommand</i>	One byte sub-command as per NEO IDG Reference Guide
<i>data</i>	Command data (if applicable)
<i>response</i>	Returns next Command response

##### Returns

RETURN\_CODE: Return codes listed as typedef enum in IDTCommon:RETURN\_CODE. Values can be parsed with IDT\_UniMagIII::device\_getResponseCodeString:()

#### 14.7.2.11 - (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

<i>val</i>	Volume level from 0.1 to 1.0
------------	------------------------------

##### 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\_UniPay::device\_getResponseCodeString:()

#### 14.7.2.12 - (RETURN\_CODE) device\_setPassThrough: (BOOL) enablePassThrough

Set Pass Through

Sets Pass-Through mode on UniPayI\_V

Parameters

<i>enablePassThrough</i>	TRUE = Pass through enabled
--------------------------	-----------------------------

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:()

#### 14.7.2.13 - (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:()

#### 14.7.2.14 - (RETURN\_CODE) emv\_authenticateTransaction: (NSData \*) tags

##### Authenticate Transaction

Authenticated a transaction after startTransaction successfully executes.

By default, auto authorize is ENABLED. If auto authorize is DISABLED, this function must be called after a result EMV\_RESULT\_CODE\_START\_TRANSACTION\_SUCCESS returned to emvTransactionData delegate protocol is received after a startTransaction call. If auto authorize is ENABLED (default), this method will automatically be executed after receiving the result EMV\_RESULT\_CODE\_START\_TRANSACTION\_SUCCESS after startTransaction. The auto authorize can be enabled/disabled with IDT\_DEVICE::disableAutoAuthenticateTransaction:()

##### Parameters

<i>tags</i>	Any tags to modify original tags submitted with startTransaction. Passed as NSData. Example, tag 9F0C with amount 0x000000000100 would be 0x9F0C06000000000100 Tag DFEE1A can be used to specify tags to be returned in response, in addition to the default tags. Example DFEE1A049F029F03 will return tags 9F02 and 9F03 with the 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:()

#### 14.7.2.15 - (RETURN\_CODE) emv\_callbackResponseLCD: (int) mode selection:(unsigned char) selection

##### Callback Response LCD Display

Provides menu selection responses to the kernel after a callback was received lcdDisplay delegate.

##### Parameters

<i>mode</i>	The choices are as follows <ul style="list-style-type: none"> <li>• 0x00 Cancel</li> <li>• 0x01 Menu Display</li> <li>• 0x02 Normal Display get Function Key supply either 0x43 ('C') for Cancel, or 0x45 ('E') for Enter/accept</li> <li>• 0x08 Language Menu Display</li> </ul>
<i>selection</i>	Line number in hex (0x01, 0x02), or 'C'/'E' of function key

## Returns

RETURN\_CODE: Values can be parsed with `errorCode.getErrorString()`

#### 14.7.2.16 - (RETURN\_CODE) emv\_completeOnlineEMVTransaction: (BOOL) *isSuccess* hostResponseTags:(NSData \*) *tags*

Complete EMV Transaction Online Request

Completes an online EMV transaction request by the card

The tags will be returned in the `emvTransactionData` delegate protocol.

## Parameters

<i>isSuccess</i>	Determines if connection to host was successful: <ul style="list-style-type: none"> <li>• TRUE: Online processing with the host (issuer) was completed</li> <li>• FALSE: Online processing could not be completed due to connection error with the host (issuer). No further data (tags) required.</li> </ul>
<i>tags</i>	Host response tag (see below)

Host response tag:

Tag	Length	Description
8A	2	Data element Authorization Response Code. Mandatory
91	8-16	Issuer Authentication Data. Optional
71	0-256	Issuer Scripts. Optional
72	0-256	Issuer Scripts. Optional
DFEE1A	0-256	Tag list of additional tags to return

Tag DFEE1A will force additional tags to be returned. Example DFEE1A049F029F03 will return tags 9F02 and 9F03 with the response

## Returns

RETURN\_CODE:

- 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
- 0xFE00: Authorization Accepted - RETURN\_CODE\_EMV\_AUTHORIZATION\_ACCEPTED
- 0xFE01: Online Failure - RETURN\_CODE\_EMV\_AUTHORIZATION\_UNABLE\_TO\_GO\_ONLINE
- 0xFE02: Technical Issue - RETURN\_CODE\_EMV\_AUTHORIZATION\_TECHNICAL\_ISSUE
- 0xFE03: Declined - RETURN\_CODE\_EMV\_AUTHORIZATION\_DECLINED
- 0xFE04: Issuer Referral - RETURN\_CODE\_EMV\_AUTHORIZATION\_ISSUER\_REFERRAL

#### 14.7.2.17 - (void) emv\_disableAutoAuthenticateTransaction: (BOOL) *disable*

Disable Auto Authenticate Transaction

If auto authenticate is DISABLED, authenticateTransaction must be called after a successful startEMV execution.

##### Parameters

<i>disable</i>	FALSE = auto authenticate ENABLED, TRUE = auto authenticate DISABLED
----------------	--

#### 14.7.2.18 - (RETURN\_CODE) emv\_getEMVL2Version: (NSString \*\*) *response*

Polls device for EMV L2 Version

##### Parameters

<i>response</i>	Response returned of Level 2 Version
-----------------	--------------------------------------

##### 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:()

#### 14.7.2.19 - (RETURN\_CODE) emv\_removeApplicationData: (NSString \*) *AID*

Remove Application Data by AID

Removes the Application Data as specified by the AID name passed as a parameter

##### Parameters

<i>AID</i>	Name of <a href="#">ApplicationID</a> in ASCII, example "A0000000031020". Must be between 5 and 16 characters
------------	---

##### 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 BTPay::device\_getResponseCodeString:()

#### 14.7.2.20 - (RETURN\_CODE) emv\_removeCAPK: (NSString \*) *rid* index:(NSString \*) *index*

Remove Certificate Authority Public Key

Removes the CAPK as specified by the RID/Index passed as a parameter in the [CAKey](#) structure

##### Parameters

<i>rid</i>	RID of the key to remove
<i>index</i>	Index of the key to remove

##### 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\_UniPayII::device\_getResponseCodeString:()

#### 14.7.2.21 - (RETURN\_CODE) emv\_removeCRLList

Remove Certificate Revocation List

Removes all [CRLEntry](#) entries

##### 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\_UniPayII::device\_getResponseCodeString:()

#### 14.7.2.22 - (RETURN\_CODE) emv\_removeTerminalData

Remove Terminal Data

Removes the Terminal Data

##### 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\_UniPayII::device\_getResponseCodeString:()

#### 14.7.2.23 - (RETURN\_CODE) emv\_retrieveAIDList: (NSArray \*\*) response

Retrieve AID list

Returns all the AID names on the terminal. Populates response parameter with an Array of NSString\* with AID names

##### Parameters

<i>response</i>	Returns a NSArray of NSString of AID Names
-----------------	--

##### 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 BTPay::device\_getResponseCodeString:()

#### 14.7.2.24 - (RETURN\_CODE) emv\_retrieveApplicationData: (NSString \*) AID response:(NSDictionary \*\*) responseAID

Retrieve Application Data by AID

- UniPay II

Retrieves the Application Data as specified by the AID name passed as a parameter. The AID will be in the response parameter responseAID

## Parameters

<i>AID</i>	Name of <a href="#">ApplicationID</a> in ASCII, example "A0000000031020". Must be between 5 and 16 characters
<i>responseAID</i>	The response returned from the method as a dictionary with Key/Object to match TagValues as follows:

Tag | Description ===== | ===== 5F57 | Account Type 9F01 | Acquirer Identifier 9F09 | Terminal application version number 5F36 | Transaction Currency Exponent 9F1B | Terminal Floor Limit 9F49 | Dynamic Data Authentication Data Object List(DDOL) 97 | Transaction Certificate Data Object List(TDOL) 9F39 | POS Entry Mode 9F3C | Transaction Reference Currency Code 9F3D | Transaction Reference Currency Exponent 99 | PIN Block DF10 | LANGUAGE DF11 | Use Trans Log DF13 | TAC-Default DF14 | TAC-Denial DF15 | TAC-Online DF17 | Threshold Value for Biased Random Selection DF18 | Target Percentage For Random Transaction Selection DF19 | Maximum Target Percentage For Random Transaction Selection DF20 | Trace DF22 | Merchant Forced Transaction Online DF25 | Default DDOL DF26 | Use Revocation list DF27 | Use Exception list DF28 | TDOL DF30 | Online DOL DF62 | Application Selection Flag DF63 | Transaction Reference Currency Conversion

## 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:()

14.7.2.25 - (RETURN\_CODE) emv\_retrieveCAPK: (NSString \*) rid index:(NSString \*) index response:(CAKey \*\*) response

Retrieve Certificate Authority Public Key

Retrieves the CAPK as specified by the RID/Index passed as a parameter in the [CAKey](#) structure. The CAPK will be in the response parameter

## Parameters

<i>rid</i>	The RID of the key to retrieve
<i>index</i>	The Index of the key to retrieve
<i>response</i>	Response returned as a <a href="#">CAKey</a>

## 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\_UniPayII::device\_getResponseCodeString:()

14.7.2.26 - (RETURN\_CODE) emv\_retrieveCAPKFile: (NSString \*) rid index:(NSString \*) index response:(NSData \*\*) response

Retrieve Certificate Authority Public Key

- BTPay 200

Retrieves the CAPK as specified by the RID/Index passed as a parameter in the [CAKey](#) structure. The CAPK will be in the response parameter

#### Parameters

<i>rid</i>	The RID of the key to retrieve
<i>index</i>	The Index of the key to retrieve
<i>response</i>	Response returned as a NSData object with the following data: <ul style="list-style-type: none"> <li>• 5 bytes RID</li> <li>• 1 byte Index</li> <li>• 1 byte Hash Algorithm</li> <li>• 1 byte Encryption Algorithm</li> <li>• 20 bytes HashValue</li> <li>• 4 bytes Public Key Exponent</li> <li>• 2 bytes Modulus Length</li> <li>• Variable bytes Modulus&gt;</li> </ul>

#### 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:()

14.7.2.27 - (RETURN\_CODE) emv\_retrieveCAPKList: (NSArray \*\*) response

Retrieve the Certificate Authority Public Key list

Returns all the CAPK RID and Index. Populates response parameter with an array of NSString items, 12 characters each, characters 1-10 RID, characters 11-12 index.

## Parameters

<i>response</i>	Response returned contains an NSArray of NSString items, 12 characters each, characters 1-10 RID, characters 11-12 index. Example "a00000000357" = RID a00000003, Index 57
-----------------	--

## 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\_UniPayII::device\_getResponseCodeString:()

14.7.2.28 - (RETURN\_CODE) emv\_retrieveCRLList: (NSMutableArray \*\*) *response*

Retrieve the Certificate Revocation List

Returns all the RID in the CRL.

## Parameters

<i>response</i>	Response returned as an NSArray of NSData objects 9-bytes each: <ul style="list-style-type: none"> <li>• 5-bytes RID, 1-byte Index, 3-byte Serial Number</li> </ul>
-----------------	---

## 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:()

14.7.2.29 - (RETURN\_CODE) emv\_retrieveTerminalData: (NSDictionary \*\*) *responseData*

Retrieve Terminal Data

- UniPay II

Retrieves the Terminal Data . The Terminal Data will be in the response parameter responseData

## Parameters

<i>responseData</i>	The response returned from the method as a dictionary with Key/Object to match TagValues as follows:
---------------------	--

9F1A | Terminal Country Code 9F35 | Terminal Type 9F33 | Terminal Capability 9F40 | Additional Terminal Capability 9F1E | IFD Serial Number 9F15 | Merchant Category Code 9F16 | Merchant Identifier 9F1C | Terminal Identification 9F4E | Merchant Name and Location DF10 | LANGUAGE DF11 | Use Trans Log DF13 | TAC-Default DF14 | TAC-Denial DF15 | TAC-Online DF17 | Threshold Value for Biased Random Selection DF18 | Target Percentage For Random Transaction Selection DF19 | Maximum Target Percentage For Random Transaction Selection DF20 | Trace DF22 | Merchant Forced Transaction Online DF25 | Default DDOL DF26 | Use Revocation list DF27 | Use Exception list DF28 | TDOL DF30 | Online DOL DF62 | Application Selection Flag DF63 | Transaction Reference Currency

## 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()

14.7.2.30 - (RETURN\_CODE) emv\_retrieveTransactionResult: (NSData \*) *tags* retrievedTags:(NSDictionary \*\*) *retrievedTags*

## Retrieve Transaction Results

Retrieves specified EMV tags from the currently executing transaction.

## Parameters

<i>tags</i>	Tags to be retrieved. Example 0x9F028A will retrieve tags 9F02 and 8A
<i>tlv</i>	All requested tags returned as unencrypted, encrypted and masked tags. The tlv NSDictionary will contain a NSDictionary with key "tags" that has the unencrypted tag data, a NSDictionary with the key "masked" that has the masked tag data, and a NSDictionary with the key "encrypted" that has the encrypted tag data

## Returns

RETURN\_CODE: Values can be parsed with device\_getResponseCodeString

14.7.2.31 - (RETURN\_CODE) emv\_setApplicationData: (NSString \*) *aidName* configData:(NSDictionary \*) *data*

## Set Application Data by AID

Sets the Application Data as specified by the dictionary containing Tags (dictionary keys) and Values (dictionary objects) according to the following table

Tag | Description ===== | ===== 5F57 | Account Type 9F01 | Acquirer Identifier 9F09 | Terminal application version number 5F36 | Transaction Currency Exponent 9F1B | Terminal Floor Limit 9F49 | Dynamic Data Authentication Data Object List(DDOL) 97 | Transaction Certificate Data Object List(TDOL) 9F39 | POS Entry Mode 9F3C | Transaction Reference Currency Code 9F3D | Transaction Reference Currency Exponent 99 | PIN Block DF10 | LANGUAGE DF11 | Use Trans Log DF13 | TAC-Default DF14 | TAC-Denial DF15 | TAC-Online DF17 | Threshold Value for Biased Random Selection DF18 | Target Percentage For Random Transaction Selection DF19 | Maximum Target Percentage For Random Transaction Selection DF20 | Trace DF22 | Merchant Forced Transaction Online DF25 | Default DDOL DF26 | Use Revocation list DF27 | Use Exception list DF28 | TDOL DF30 | Online DOL DF62 | Application Selection Flag DF34 | Authorization Response Code: 1-2 bytes=approved code, 3-4 bytes = referral code, 5-6 bytes = declined code

#### Parameters

<i>aidName</i>	aidName AID name. Example "a0000000031010"
<i>data</i>	NSDictionary with Tags/Values for the AID configuration file

#### 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:()

#### 14.7.2.32 - (RETURN\_CODE) emv\_setCAPK: (CAKey) key

Set Certificate Authority Public Key

Sets the CAPK as specified by the [CAKey](#) structure

#### Parameters

<i>key</i>	<a href="#">CAKey</a> containing the RID, Index, and key data to set
------------	--

#### 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\_UniPayII::device\_getResponseCodeString:()



## 14.7.2.33 - (RETURN\_CODE) emv\_setCAPKFile: (NSData \*) file

Set Certificate Authority Public Key

Sets the CAPK as specified by the [CAKey](#) raw format

## Parameters

key	<p><a href="#">CAKey</a> format: [5 bytes RID][1 byte Index][1 byte Hash Algorithm][1 byte Encryption Algorithm][20 bytes HashValue][4 bytes Public Key Exponent][2 bytes Modulus Length][Variable bytes Modulus] Where:</p> <ul style="list-style-type: none"> <li>• Hash Algorithm: The only algorithm supported is SHA-1. The value is set to 0x01</li> <li>• Encryption Algorithm: The encryption algorithm in which this key is used. Currently support only one type: RSA. The value is set to 0x01.</li> <li>• HashValue: Which is calculated using SHA-1 over the following fields: RID &amp; Index &amp; Modulus &amp; Exponent</li> <li>• Public Key Exponent: Actually, the real length of the exponent is either one byte or 3 bytes. It can have two values: 3 (Format is 0x00 00 00 03), or 65537 (Format is 0x00 01 00 01)</li> <li>• Modulus Length: LenL LenH Indicated the length of the next field.</li> <li>• Modulus: This is the modulus field of the public key. Its length is specified in the field above.</li> </ul>
-----	--

## Returns

RETURN\_CODE: Values can be parsed with `errorCode.getErrorString()`

## 14.7.2.34 - (RETURN\_CODE) emv\_setCRLentries: (NSData \*) data

Set Certificate Revocation List

Sets the CRL list

## Parameters

data	<p>CRLentries as a repeating occurrence of CRL: CRL1 CRL2 ... CRLn. CRL format is</p> <ul style="list-style-type: none"> <li>• 5Bytes RID</li> <li>• 1Byte CA public key Index</li> <li>• 3Bytes Certificate Serial Number</li> </ul>
------	---

## 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\_UniPayII::device\_getResponseCodeString:()

#### 14.7.2.35 - (RETURN\_CODE) emv\_setTerminalData: (NSDictionary \*) data

##### Set Terminal Data

- UniPay II

Sets the Terminal Data as specified by the dictionary containing Tags (dictionary keys) and Values (dictionary objects) according to the following table

Tag | Description ===== | ===== 5F36 | Transaction Currency Exponent 9F1A | Terminal Country Code 9F35 | Terminal Type 9F33 | Terminal Capability 9F40 | Additional Terminal Capability 9F1E | IFD Serial Number 9F15 | Merchant Category Code 9F16 | Merchant Identifier 9F1C | Terminal Identification 9F4E | Merchant Name and Location DF10 | LANGUAGE DF11 | Use Trans Log DFEE13 | TAC-Default DFEE14 | TAC-Denial DFEE2A | Threshold Value for Biased Random Selection DF18 | Target Percentage For Random Transaction Selection DF19 | Maximum Target Percentage For Random Transaction Selection DF20 | Trace DF22 | Merchant Forced Transaction Online DF25 | Default DDOL DF26 | Use Revocation list DF27 | Use Exception list DF28 | TDOL DF30 | Online DOL DFEE15 | Application Selection Flag DF63 | Transaction Reference Currency Conversion

##### Parameters

<i>data</i>	NSDictionary with Tags/Values for the Terminal configuration file
-------------	---

##### 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:()

#### 14.7.2.36 - (RETURN\_CODE) emv\_startTransaction: (double) amount amtOther:(double) amtOther type:(int) type timeout:(int) timeout tags:(NSData \*) tags forceOnline:(BOOL) forceOnline fallback:(BOOL) fallback

##### Start EMV Transaction Request

Authorizes the EMV transaction for an ICC card

The tags will be returned in the emvTransactionData delegate protocol.

By default, auto authorize is ENABLED. If auto authorize is DISABLED, this function will complete with a return of EMV\_RESULT\_CODE\_START\_TRANSACTION\_SUCCESS to emvTransactionData delegate protocol, and then IDT\_UniPayII::emv\_authenticateTransaction() must be executed. If auto authorize is ENABLED (default), IDT\_UniPayII::emv\_authenticateTransaction() will automatically be executed after receiving the result EMV\_RESULT\_CODE\_START\_TRANSACTION\_SUCCESS. The auto authorize can be enabled/disabled with [emv\\_disableAutoAuthenticateTransaction:](#)

## Parameters

<i>amount</i>	Transaction amount value (tag value 9F02)
<i>amtOther</i>	Other amount value, if any (tag value 9F03)
<i>type</i>	Transaction type (tag value 9C).
<i>timeout</i>	Timeout value in seconds.
<i>tags</i>	Any other tags to be included in the request. Passed as a NSData. Example, tag 9F0C with amount 0x000000000100 would be 0x9F0C06000000000100 If tags 9F02 (amount), 9F03 (other amount), or 9C (transaction type) are included, they will take priority over these values supplied as individual parameters to this method. Tag DFEE1A can be used to specify tags to be returned in response, in addition to the default tags. Example DFEE1A049F029F03 will return tags 9F02 and 9F03 with the response
<i>forceOnline</i>	TRUE = do not allow offline approval, FALSE = allow ICC to approve offline if terminal capable
<i>autoAuthenticate</i>	Will automatically execute Authenticate Transaction after start transaction returns successful
<i>fallback</i>	Indicate if it supports fallback to MSR

## Returns

RETURN\_CODE: Values can be parsed with `errorCode.getErrorString()`

14.7.2.37 - (RETURN\_CODE) `icc_exchangeAPDU: (NSData *) dataAPDU response:(APDUResponse **) response`

Exchange APDU (unencrypted)

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

## Parameters

<i>dataAPDU</i>	APDU data packet
<i>response</i>	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](#):

14.7.2.38 - (RETURN\_CODE) `icc_getICCReaderStatus: (ICCReaderStatus **) readerStatus`

Get Reader Status

Returns the reader status

## Parameters

<i>readerStatus</i>	Pointer that will return with the <a href="#">ICCReaderStatus</a> 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_Device sharedController] getICCReaderStatus:&readerStatus];
if (RETURN_CODE_DO_SUCCESS != rt) {
    NSLog(@"Fail");
}
else{
    NSString *sta;
    if (readerStatus->iccPower)
        sta=@"[ICC Powered]";
    else
        sta=@"[ICC Power not Ready]";
    if (readerStatus->cardSeated)
        sta=[NSString stringWithFormat:@"%s",[Card Seated]", sta];
    else
        sta=[NSString stringWithFormat:@"%s",[Card not Seated]", sta];
}

```

#### 14.7.2.39 - (RETURN\_CODE) icc\_powerOffICC: (NSString \*\*) *error*

Power Off ICC

Powers down the ICC

##### Parameters

<i>error</i>	Returns the error, if any
--------------	---------------------------

##### Returns

RETURN\_CODE: Return codes listed as typedef enum in IDTCommon:RETURN\_CODE.

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

#### 14.7.2.40 - (RETURN\_CODE) icc\_powerOnICC: (NSData \*\*) *response*

Power On ICC

Power up the currently selected microprocessor card in the ICC reader

##### Parameters

<i>response</i>	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.

#### 14.7.2.41 - (bool) isConnected

Check if device is connected

#### 14.7.2.42 - (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 IDT\_UniPay::device\_getResponseCodeString:()

**14.7.2.43 - (RETURN\_CODE) msr\_startMSRSwipe****Enable MSR Swipe**

Enables CLTS and MSR, waiting for swipe or tap to occur. If swipe captured, returns [IDTMSRData](#) instance to deviceDelegate::swipeMSRData:(). If CTLS captured, returns [IDTEMVData](#) to deviceDelegate::emvTransaction←Data:()

**Returns**

RETURN\_CODE: Return codes listed as typedef enum in IDTCommon:RETURN\_CODE. Values can be parsed with IDT\_UniPay::device\_getResponseCodeString:()

**14.7.2.44 + (NSString\*) SDK\_version****SDK Version**

Returns the current version of IDTech.framework

**Returns**

Framework version

**14.7.2.45 + (IDT\_UniPayI\_V\*) sharedController****Singleton Instance**

Establishes an singleton instance of [IDT\\_UniPayI\\_V](#) class.

**Returns**

Instance of [IDT\\_UniPayI\\_V](#)

**14.7.3 Property Documentation****14.7.3.1 - (id< IDT\_UniPayI\_V\_Delegate >) delegate [read],[write],[atomic],[strong]**

- Reference to [IDT\\_UniPayI\\_V\\_Delegate](#).

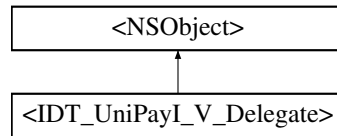
The documentation for this class was generated from the following file:

- Source\_iOS/IDT\_UniPayI\_V.h

**14.8 <IDT\_UniPayI\_V\_Delegate> Protocol Reference**

```
#import <IDT_UniPayI_V.h>
```

Inheritance diagram for <IDT\_UniPayI\_V\_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) - [lcdDisplay:lines:](#)
- (void) - [emvTransactionData:errorCode:](#)

### 14.8.1 Detailed Description

Protocol methods established for [IDT\\_UniPayI\\_V](#) class

### 14.8.2 Method Documentation

14.8.2.1 - (void) [dataInOutMonitor:](#) (NSData \*) *data* [incoming:\(BOOL\) isIncoming](#) [optional]

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

#### Parameters

<i>data</i>	The serial data represented as a NSData object
<i>isIncoming</i>	The direction of the data <ul style="list-style-type: none"> <li>• TRUE specifies data being received from the device,</li> <li>• FALSE indicates data being sent to the device.</li> </ul>

14.8.2.2 - (void) [deviceMessage:](#) (NSString \*) *message* [optional]

Receives messages from the framework

#### Parameters

<i>message</i>	String message transmitted by framework
----------------	---

14.8.2.3 - (void) [emvTransactionData:](#) (IDTEMVData \*) *emvData* [errorCode:\(int\) error](#) [optional]

EMV Transaction Data

This protocol will receive results from IDT\_Device::startEMVTransaction:otherAmount:timeout:cashback↵:additionalTags:()

#### Parameters

<i>emvData</i>	EMV Results Data. Result code, card type, encryption type, masked tags, encrypted tags, unencrypted tags and KSN
<i>error</i>	The error code as defined in the errors.h file

#### 14.8.2.4 - (void) lcdDisplay: (int) *mode* lines:(NSArray \*) *lines* [optional]

LCD Display Request During an EMV transaction, this delegate will receive data to clear virtual LCD display, display messages, display menu, or display language. Applies to UniPay III

#### Parameters

<i>mode</i>	LCD Display Mode: <ul style="list-style-type: none"> <li>• 0x01: Menu Display. A selection must be made to resume the transaction</li> <li>• 0x02: Normal Display get function key. A function must be selected to resume the transaction</li> <li>• 0x03: Display without input. Message is displayed without pausing the transaction</li> <li>• 0x04: List of languages are presented for selection. A selection must be made to resume the transaction</li> <li>• 0x10: Clear Screen. Command to clear the LCD screen</li> </ul>
-------------	---

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

Monitors the headphone jack for device insertion/removal.

#### Parameters

<i>deviceInserted</i>	TRUE = device inserted, FALSE = device removed
-----------------------	--

#### 14.8.2.6 - (void) swipeMSRData: (IDTMSRData \*) *cardData* [optional]

Receives card data from MSR swipe.

#### Parameters

<i>cardData</i>	Captured card data from MSR swipe
-----------------	-----------------------------------

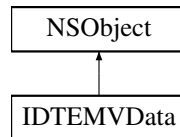
The documentation for this protocol was generated from the following file:

- Source\_iOS/IDT\_UniPayI\_V.h

## 14.9 IDTEMVData Class Reference

```
#import <IDTEMVData.h>
```

Inheritance diagram for IDTEMVData:



### Instance Methods

- (void) - [clear](#)

### Class Methods

- ([IDTEMVData \\*](#)) + [sharedController](#)

### Properties

- EMV\_RESULT\_CODE\_Types [resultCode](#)
- EMV\_RESULT\_CODE\_V2\_Types [resultCodeV2](#)
- int [encryptionMode](#)  
0 = *TDES*, 1 = *AES*
- int [cardType](#)  
0 = *Contact*, 1 = *Contactless*
- bool [hasAdvise](#)  
*TRUE if response has an Advise request.*
- bool [hasReversal](#)  
*TRUE if response has reversal request.*
- NSDictionary \* [unencryptedTags](#)  
*Unencrypted EMV Tags. Key = tag name (NSString), Object = tag value (NSData)*
- NSDictionary \* [encryptedTags](#)  
*Encrypted EMV Tags. Key = tag name (NSString), Object = tag value (NSData)*
- NSDictionary \* [maskedTags](#)  
*Encrypted EMV Tags. Key = tag name (NSString), Object = tag value (NSData)*
- NSData \* [KSN](#)  
*Key Serial Number for encrypted EMV tags.*
- [IDTMSRData \\*](#) [cardData](#)  
*Card data returned from fallback or non-icc swipe during emv transaction.*

#### 14.9.1 Detailed Description

Encapsulating data class for EMV data capture

#### 14.9.2 Method Documentation

##### 14.9.2.1 - (void) clear

clears all [IDTEMVData](#) properties



### 14.9.2.2 + (IDTEMVData \*) sharedController

Singleton instance of [IDTEMVData](#)

## 14.9.3 Property Documentation

### 14.9.3.1 - (EMV\_RESULT\_CODE\_Types) resultCode [read],[write],[atomic]

Result Code. Uses enumeration EMV\_RESULT\_CODE\_Types

```
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_Types;
```

### 14.9.3.2 - (EMV\_RESULT\_CODE\_V2\_Types) resultCodeV2 [read],[write],[atomic]

Result Code. Uses enumeration EMV\_RESULT\_CODE\_V2\_Types

```
typedef enum{
    EMV_RESULT_CODE_V2_APPROVED_OFFLINE = 0x0000,
    EMV_RESULT_CODE_V2_DECLINED_OFFLINE = 0x0001,
    EMV_RESULT_CODE_V2_APPROVED = 0x0002,
    EMV_RESULT_CODE_V2_DECLINED = 0x0003,
    EMV_RESULT_CODE_V2_GO_ONLINE = 0x0004,
    EMV_RESULT_CODE_V2_CALL_YOUR_BANK = 0x0005,
    EMV_RESULT_CODE_V2_NOT_ACCEPTED = 0x0006,
    EMV_RESULT_CODE_V2_USE_MAGSTRIPE = 0x0007,
    EMV_RESULT_CODE_V2_TIME_OUT = 0x0008,
    EMV_RESULT_CODE_V2_START_TRANS_SUCCESS = 0x0010,
    EMV_RESULT_CODE_V2_MSR_SUCCESS = 0x0011,
    EMV_RESULT_CODE_V2_FILE_ARG_INVALID = 0x1001,
    EMV_RESULT_CODE_V2_FILE_OPEN_FAILED = 0x1002,
    EMV_RESULT_CODE_V2_FILE_OPERATION_FAILED = 0x1003,
    EMV_RESULT_CODE_V2_MEMORY_NOT_ENOUGH = 0x2001,
    EMV_RESULT_CODE_V2_SMARTCARD_FAIL = 0x3001,
    EMV_RESULT_CODE_V2_SMARTCARD_INIT_FAILED = 0x3003,
    EMV_RESULT_CODE_V2_FALLBACK_SITUATION = 0x3004,
    EMV_RESULT_CODE_V2_SMARTCARD_ABSENT = 0x3005,
    EMV_RESULT_CODE_V2_SMARTCARD_TIMEOUT = 0x3006,
    EMV_RESULT_CODE_V2_MSR_CARD_ERROR = 0x3007,
    EMV_RESULT_CODE_V2_PARSING_TAGS_FAILED = 0x5001,
    EMV_RESULT_CODE_V2_CARD_DATA_ELEMENT_DUPLICATE = 0x5002,
    EMV_RESULT_CODE_V2_DATA_FORMAT_INCORRECT = 0x5003,
    EMV_RESULT_CODE_V2_APP_NO_TERM = 0x5004,
    EMV_RESULT_CODE_V2_APP_NO_MATCHING = 0x5005,
    EMV_RESULT_CODE_V2_AMANDATORY_OBJECT_MISSING = 0x5006,
    EMV_RESULT_CODE_V2_APP_SELECTION_RETRY = 0x5007,
    EMV_RESULT_CODE_V2_AMOUNT_ERROR_GET = 0x5008,
    EMV_RESULT_CODE_V2_CARD_REJECTED = 0x5009,
    EMV_RESULT_CODE_V2_AIP_NOT_RECEIVED = 0x5010,
    EMV_RESULT_CODE_V2_AFL_NOT_RECEIVEDE = 0x5011,
    EMV_RESULT_CODE_V2_AFL_LEN_OUT_OF_RANGE = 0x5012,
    EMV_RESULT_CODE_V2_SFI_OUT_OF_RANGE = 0x5013,
    EMV_RESULT_CODE_V2_AFL_INCORRECT = 0x5014,
```

```

EMV_RESULT_CODE_V2_EXP_DATE_INCORRECT = 0x5015,
EMV_RESULT_CODE_V2_EFF_DATE_INCORRECT = 0x5016,
EMV_RESULT_CODE_V2_ISS_COD_TBL_OUT_OF_RANGE = 0x5017,
EMV_RESULT_CODE_V2_CRYPTOGAM_TYPE_INCORRECT = 0x5018,
EMV_RESULT_CODE_V2_PSE_BY_CARD_NOT_SUPPORTED = 0x5019,
EMV_RESULT_CODE_V2_USER_LANGUAGE_SELECTED = 0x5020,
EMV_RESULT_CODE_V2_SERVICE_NOT_ALLOWED = 0x5021,
EMV_RESULT_CODE_V2_NO_TAG_FOUND = 0x5022,
EMV_RESULT_CODE_V2_CARD_BLOCKED = 0x5023,
EMV_RESULT_CODE_V2_LEN_INCORRECT = 0x5024,
EMV_RESULT_CODE_V2_CARD_COM_ERROR = 0x5025,
EMV_RESULT_CODE_V2_TSC_NOT_INCREASED = 0x5026,
EMV_RESULT_CODE_V2_HASH_INCORRECT = 0x5027,
EMV_RESULT_CODE_V2_ARC_NOT_PRESENCE = 0x5028,
EMV_RESULT_CODE_V2_ARC_INVALID = 0x5029,
EMV_RESULT_CODE_V2_COMM_NO_ONLINE = 0x5030,
EMV_RESULT_CODE_V2_TRAN_TYPE_INCORRECT = 0x5031,
EMV_RESULT_CODE_V2_APP_NO_SUPPORT = 0x5032,
EMV_RESULT_CODE_V2_APP_NOT_SELECT = 0x5033,
EMV_RESULT_CODE_V2_LANG_NOT_SELECT = 0x5034,
EMV_RESULT_CODE_V2_TERM_DATA_NOT_PRESENCE = 0x5035,
EMV_RESULT_CODE_V2_CVM_TYPE_UNKNOWN = 0x6001,
EMV_RESULT_CODE_V2_CVM_AIP_NOT_SUPPORTED = 0x6002,
EMV_RESULT_CODE_V2_CVM_TAG_8E_MISSING = 0x6003,
EMV_RESULT_CODE_V2_CVM_TAG_8E_FORMAT_ERROR = 0x6004,
EMV_RESULT_CODE_V2_CVM_CODE_IS_NOT_SUPPORTED = 0x6005,
EMV_RESULT_CODE_V2_CVM_COND_CODE_IS_NOT_SUPPORTED = 0x6006,
EMV_RESULT_CODE_V2_CVM_NO_MORE = 0x6007,
EMV_RESULT_CODE_V2_PIN_BYPASSED_BEFORE = 0x6008
} EMV_RESULT_CODE_V2_Types;

```

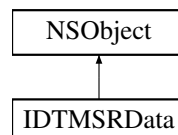
The documentation for this class was generated from the following file:

- [Source\\_iOS/IDTEMVData.h](#)

## 14.10 IDTMSRData Class Reference

```
#import <IDTMSRData.h>
```

Inheritance diagram for IDTMSRData:



### Instance Methods

- (void) - [clear](#)

### Class Methods

- (IDTMSRData \*) + [sharedController](#)

### Properties

- EVENT\_MSR\_Types [event](#)
- CAPTURE\_ENCODE\_TYPE [captureEncodeType](#)
- CAPTURE\_ENCRYPT\_TYPE [captureEncryptType](#)
- NSData \* [cardData](#)  
*Complete unparsed swipe data as received from MSR.*
- NSString \* [track1](#)

- Track 1 masked if encryption enabled or cleartext if encryption disabled.*

  - NSString \* [track2](#)

*Track 2 masked if encryption enabled or cleartext if encryption disabled.*

  - NSString \* [track3](#)

*Track 3 masked if encryption enabled or cleartext if encryption disabled.*

  - int [track1Length](#)

*Length of track 1 masked/clear text data.*

  - int [track2Length](#)

*Length of track 2 masked/clear text data.*

  - int [track3Length](#)

*Length of track 3 masked/clear text data.*

  - NSData \* [encTrack1](#)

*Track 1 encoded data OR all encoded track data if encryption method combines all tracks into single blob.*

  - NSData \* [encTrack2](#)

*Track 2 encoded.*

  - NSData \* [encTrack3](#)

*Track 3 encoded.*

  - NSData \* [hashTrack1](#)

*Sha-256 hash of Track 1 encoded data.*

  - NSData \* [hashTrack2](#)

*Sha-256 hash of Track 2 encoded data.*

  - NSData \* [hashTrack3](#)

*Sha-256 hash of Track 3 encoded data.*

  - NSString \* [RSN](#)

*Reader Serial Number.*

  - NSData \* [KSN](#)

*Key Serial Number.*

  - NSData \* [sessionID](#)

*Session ID - Security level 4 only.*

  - unsigned char [readStatus](#)
  - int [errorCode](#)

*Contains error code when data is not returned.*

  - bool [iccPresent](#)

*Card contains ICC.*

  - NSDictionary \* [unencryptedTags](#)

*Unencrypted card data provided via TLV.*

  - NSDictionary \* [encryptedTags](#)

*Encrypted card data provided via TLV.*

  - NSDictionary \* [maskedTags](#)

*Masked card data provided via TLV.*

### 14.10.1 Detailed Description

Encapsulating data class for MSR data capture

### 14.10.2 Method Documentation

#### 14.10.2.1 - (void) clear

clears all [IDTMSRData](#) properties

#### 14.10.2.2 + (IDTMSRData \*) sharedController

Singleton instance of [IDTMSRData](#)

### 14.10.3 Property Documentation

#### 14.10.3.1 - (CAPTURE\_ENCODE\_TYPE) captureEncodeType [read],[write],[atomic]

Encode Type of captured MSR Data.

Uses enumeration CAPTURE\_ENCODE\_TYPE:

```
typedef enum{
    CAPTURE_ENCODE_TYPE_ISOABA=0,
    CAPTURE_ENCODE_TYPE_AAMVA=1,
    CAPTURE_ENCODE_TYPE_Other=3,
    CAPTURE_ENCODE_TYPE_Raw=4
} CAPTURE_ENCODE_TYPE;
```

#### 14.10.3.2 - (CAPTURE\_ENCRYPT\_TYPE) captureEncryptType [read],[write],[atomic]

Encrypt Type of captured MSR Data.

Uses enumeration CAPTURE\_ENCODE\_TYPE:

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

#### 14.10.3.3 - (EVENT\_MSR\_Types) event [read],[write],[atomic]

Event type. Uses enumeration EVENT\_MSR\_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_UNIPAY_KEYLOADING,
    EVENT_MSR_TIMEOUT
}EVENT_MSR_Types;
```

#### 14.10.3.4 - (unsigned char) readStatus [read],[write],[atomic]

Track Read Status

- Bit 0: 1=Track 1 decode success, 0=Track 1 decode fail
- Bit 1: 1=Track 2 decode success, 0=Track 2 decode fail)
- Bit 2: 1=Track 3 decode success, 0=Track 3 decode fail)
- Bit 3: 1=Track 1 sampling data exists, 0=Track 1 sampling data does not exist
- Bit 4: 1=Track 2 sampling data exists, 0=Track 2 sampling data does not exist

- Bit 5: 1=Track 3 sampling data exists, 0=Track 3 sampling data does not exist
- Bit 6: reserved for future use
- Bit 7: reserved for future use

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

- Source\_iOS/IDTMSRData.h

## 14.11 MaskAndEncryption Struct Reference

```
#include <IDTCommon.h>
```

### Public Attributes

- unsigned char [prePANClear](#)  
*Leading PAN digits to display. Values '0' - '6'. Default '4'.*
- unsigned char [postPANClear](#)  
*Last PAN digits to display. Values '0' - '4'. Default '4'.*
- unsigned char [maskChar](#)  
*Last PAN digits to display. Values 0x20-0x7E. Default 0x2A '\*'.*
- unsigned char [displayExpDate](#)  
*Mask or display expiration date. Values '0' = mask, '1' = don't mask. Default '1'.*
- unsigned char [baseKeyType](#)  
*BTPay Only. Key Type. Values '0' = Data Key, '1' = Pin Key. Default '0'.*
- unsigned char [encryptionType](#)  
*BTPay Only. Key Type. Values '1' = TDES, '2' = AES. Default '1'.*
- unsigned char [encryptionOption](#)
- unsigned char [maskOption](#)

### 14.11.1 Detailed Description

Mask and Encryption - Used to Set/Retrieve mask and encryption values IDT\_BTPay::emv\_retrieveAIDList() IDT↔\_UniPay::emv\_retrieveAIDList().

### 14.11.2 Member Data Documentation

#### 14.11.2.1 unsigned char MaskAndEncryption::encryptionOption

UniPay II Only. Bit 0: T1 force encrypt Bit 1 : T2 force encrypt Bit 2 : T3 force encrypt Bit3 : T3 force encrypt when card type is 0

#### 14.11.2.2 unsigned char MaskAndEncryption::maskOption

UniPay II Only. Bit0: T1 mask allowed Bit1: T2 mask allowed Bit2: T3 mask allowed

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

- Source\_iOS/IDTCommon.h

## 14.12 PowerOnStructure Struct Reference

```
#include <IDTCommon.h>
```

### Public Attributes

- 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](#)  
*length of pps data*

### 14.12.1 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:

- Source\_iOS/IDTCommon.h

## 14.13 TerminalData Struct Reference

```
#include <IDTCommon.h>
```

### Public Attributes

- unsigned char [terminalCountryCode](#) [2]  
*Terminal's location. Tag 9F1A {0x08,0x40}.*
- unsigned char [provideCardholderConfirmation](#)  
*Indicates whether or not cardholder may confirm application selection at EMV Selection time. Tag 58 0x00 or 0x01.*
- unsigned char [terminalType](#)  
*Standard parameter. Tag 9F35 See EMVCo book IV.*
- unsigned char [emvContact](#)  
*Indicates whether terminal supports EMV contact. Tag 9F33, byte 1, bit 6 0x00 or 0x01.*
- unsigned char [terminalCapabilities](#) [3]  
*Standard parameter. Tag 9F33 See EMVCo book IV.*
- unsigned char [additionalTerminalCapabilities](#) [5]  
*Standard parameter. Tag 9F40 See EMVCo book IV.*
- unsigned char [emvContactless](#)  
*Indicates whether or not terminal support scontactless in EMV mode. 0x00 or 0x01.*

- unsigned char [magstripe](#)  
*Indicates whether terminal supports magstripe. 0x00 or 0x01.*
- unsigned char [pinTimeOut](#)  
*In seconds. Time allocated to cardholder to enter PIN. Binary value Example : 0x0F for 15s.*
- unsigned char [batchManaged](#)  
*Indicates whether or not Batch messages are supported by Terminal. 0x00 or 0x01.*
- unsigned char [adviceManaged](#)  
*Indicates whether or not Advice messages are supported by Terminal (only if needed by Level3 implementation). 0x00 or 0x01.*
- unsigned char [pse](#)  
*Indicates whether or not PSE Selection method is supported by Terminal. 0x00 or 0x01.*
- unsigned char [autoRun](#)  
*Indicates whether or not Terminal is configured in AutoRun. 0x00 or 0x01.*
- unsigned char [predefinedAmount](#) [3]  
*Fixed amount. Binary value.*
- unsigned char [pinByPass](#)  
*Indicates whether or not PIN bypass is supported by Terminal. 0x00 or 0x01.*
- unsigned char [referralManaged](#)  
*Indicates whether or not Referral managed are supported by Terminal (only if needed by Level3 implementation).. 0x00 or 0x01.*
- unsigned char [defaultTAC](#)  
*Indicates whether or not Default TAC are supported by Terminal. 0x00 or 0x01.*
- unsigned char [defaultTACDenial](#) [5]  
*Default TAC Denial value. See EMVCo book IV.*
- unsigned char [defaultTACOnline](#) [5]  
*Default TAC Online value. See EMVCo book IV.*
- unsigned char [defaultTACDefault](#) [5]  
*Default TAC Default value. See EMVCo book IV.*
- unsigned char [notRTS](#)  
*Indicates RTS are not supported by Terminal or not. 0x00 or 0x01.*
- unsigned char [notVelocity](#)  
*Indicates Velocity are not supported by Terminal or not. 0x00 or 0x01.*
- unsigned char [cdaType](#)  
*Supported CDA type. Value should be 0x02.*

### 14.13.1 Detailed Description

device Terminal Configuration File - 44 bytes

Used as parameter in `IDT_BTPay::setTerminalData:()`

Used as return value in `IDT_BTPay::emv_retrieveTerminalData:()`

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

- `Source_iOS/IDTCommon.h`





# Index

<IDT\_UniPayI\_V\_Delegate>, [101](#)

AIDEntry, [76](#)

APDUResponse, [76](#)

clear, [77](#)

sharedController, [77](#)

ApplicationID, [77](#)

CAKey, [79](#)

CRLEntry, [79](#)

captureEncodeType

IDTMSRData, [108](#)

captureEncryptType

IDTMSRData, [108](#)

clear

APDUResponse, [77](#)

IDTEMVData, [104](#)

IDTMSRData, [107](#)

close

IDT\_UniPayI\_V, [82](#)

config\_getSerialNumber:

IDT\_UniPayI\_V, [82](#)

dataInOutMonitor:incoming:

IDT\_UniPayI\_V\_Delegate-p, [102](#)

delegate

IDT\_UniPayI\_V, [101](#)

device\_cancelConnectToAudioReader

IDT\_UniPayI\_V, [82](#)

device\_connectToAudioReader

IDT\_UniPayI\_V, [82](#)

device\_connectToUSB

IDT\_UniPayI\_V, [82](#)

device\_getFirmwareVersion:

IDT\_UniPayI\_V, [83](#)

device\_getResponseCodeString:

IDT\_UniPayI\_V, [83](#)

device\_isAudioReaderConnected

IDT\_UniPayI\_V, [83](#)

device\_isConnected:

IDT\_UniPayI\_V, [83](#)

device\_sendIDGCommand:subCommand:data↔

:response:

IDT\_UniPayI\_V, [84](#)

device\_setAudioVolume:

IDT\_UniPayI\_V, [84](#)

device\_setPassThrough:

IDT\_UniPayI\_V, [85](#)

device\_startRKI

IDT\_UniPayI\_V, [85](#)

deviceMessage:

IDT\_UniPayI\_V\_Delegate-p, [102](#)

emv\_authenticateTransaction:

IDT\_UniPayI\_V, [85](#)

emv\_callbackResponseLCD:selection:

IDT\_UniPayI\_V, [86](#)

emv\_completeOnlineEMVTransaction:hostResponse↔

Tags:

IDT\_UniPayI\_V, [87](#)

emv\_disableAutoAuthenticateTransaction:

IDT\_UniPayI\_V, [87](#)

emv\_getEMVL2Version:

IDT\_UniPayI\_V, [88](#)

emv\_removeApplicationData:

IDT\_UniPayI\_V, [88](#)

emv\_removeCAPK:index:

IDT\_UniPayI\_V, [89](#)

emv\_removeCRLList

IDT\_UniPayI\_V, [89](#)

emv\_removeTerminalData

IDT\_UniPayI\_V, [89](#)

emv\_retrieveAIDList:

IDT\_UniPayI\_V, [90](#)

emv\_retrieveApplicationData:response:

IDT\_UniPayI\_V, [90](#)

emv\_retrieveCAPK:index:response:

IDT\_UniPayI\_V, [91](#)

emv\_retrieveCAPKFile:index:response:

IDT\_UniPayI\_V, [92](#)

emv\_retrieveCAPKList:

IDT\_UniPayI\_V, [92](#)

emv\_retrieveCRLList:

IDT\_UniPayI\_V, [94](#)

emv\_retrieveTerminalData:

IDT\_UniPayI\_V, [94](#)

emv\_retrieveTransactionResult:retrievedTags:

IDT\_UniPayI\_V, [95](#)

emv\_setApplicationData:configData:

IDT\_UniPayI\_V, [95](#)

emv\_setCAPK:

IDT\_UniPayI\_V, [96](#)

emv\_setCAPKFile:

IDT\_UniPayI\_V, [96](#)

emv\_setCRLEntries:

IDT\_UniPayI\_V, [97](#)

emv\_setTerminalData:

IDT\_UniPayI\_V, [98](#)

emv\_startTransaction:amtOther:type:timeout:tags↔

:forceOnline:fallback:

- IDT\_UniPayI\_V, 98
- emvTransactionData:errorCode:
  - IDT\_UniPayI\_V\_Delegate-p, 102
- encryptionOption
  - MaskAndEncryption, 109
- event
  - IDTMSRData, 108
- ICCReaderStatus, 80
- IDT\_UniPayI\_V\_Delegate-p
  - dataInOutMonitor:incoming:, 102
  - deviceMessage:, 102
  - emvTransactionData:errorCode:, 102
  - lcdDisplay:lines:, 103
  - plugStatusChange:, 103
  - swipeMSRData:, 103
- IDT\_UniPayI\_V, 81
  - close, 82
  - config\_getSerialNumber:, 82
  - delegate, 101
  - device\_cancelConnectToAudioReader, 82
  - device\_connectToAudioReader, 82
  - device\_connectToUSB, 82
  - device\_getFirmwareVersion:, 83
  - device\_getResponseCodeString:, 83
  - device\_isAudioReaderConnected, 83
  - device\_isConnected:, 83
  - device\_sendIDGCommand:subCommand:data↔:response:, 84
  - device\_setAudioVolume:, 84
  - device\_setPassThrough:, 85
  - device\_startRKI, 85
  - emv\_authenticateTransaction:, 85
  - emv\_callbackResponseLCD:selection:, 86
  - emv\_completeOnlineEMVTransaction:host↔ResponseTags:, 87
  - emv\_disableAutoAuthenticateTransaction:, 87
  - emv\_getEMVL2Version:, 88
  - emv\_removeApplicationData:, 88
  - emv\_removeCAPK:index:, 89
  - emv\_removeCRLList, 89
  - emv\_removeTerminalData, 89
  - emv\_retrieveAIDList:, 90
  - emv\_retrieveApplicationData:response:, 90
  - emv\_retrieveCAPK:index:response:, 91
  - emv\_retrieveCAPKFile:index:response:, 92
  - emv\_retrieveCAPKList:, 92
  - emv\_retrieveCRLList:, 94
  - emv\_retrieveTerminalData:, 94
  - emv\_retrieveTransactionResult:retrievedTags:, 95
  - emv\_setApplicationData:configData:, 95
  - emv\_setCAPK:, 96
  - emv\_setCAPKFile:, 96
  - emv\_setCRLEntries:, 97
  - emv\_setTerminalData:, 98
  - emv\_startTransaction:amtOther:type:timeout↔:tags:forceOnline:fallback:, 98
  - icc\_exchangeAPDU:response:, 99
  - icc\_getICCReaderStatus:, 99
- icc\_powerOffICC:, 100
- icc\_powerOnICC:, 100
- isConnected, 100
- msr\_cancelMSRSwipe, 100
- msr\_startMSRSwipe, 101
- SDK\_version, 101
- sharedController, 101
- IDTEMVData, 104
  - clear, 104
  - resultCode, 105
  - resultCodeV2, 105
  - sharedController, 104
- IDTMSRData, 106
  - captureEncodeType, 108
  - captureEncryptType, 108
  - clear, 107
  - event, 108
  - readStatus, 108
  - sharedController, 107
- icc\_exchangeAPDU:response:
  - IDT\_UniPayI\_V, 99
- icc\_getICCReaderStatus:
  - IDT\_UniPayI\_V, 99
- icc\_powerOffICC:
  - IDT\_UniPayI\_V, 100
- icc\_powerOnICC:
  - IDT\_UniPayI\_V, 100
- isConnected
  - IDT\_UniPayI\_V, 100
- lcdDisplay:lines:
  - IDT\_UniPayI\_V\_Delegate-p, 103
- MaskAndEncryption, 109
  - encryptionOption, 109
  - maskOption, 109
- maskOption
  - MaskAndEncryption, 109
- msr\_cancelMSRSwipe
  - IDT\_UniPayI\_V, 100
- msr\_startMSRSwipe
  - IDT\_UniPayI\_V, 101
- plugStatusChange:
  - IDT\_UniPayI\_V\_Delegate-p, 103
- PowerOnStructure, 110
- readStatus
  - IDTMSRData, 108
- resultCode
  - IDTEMVData, 105
- resultCodeV2
  - IDTEMVData, 105
- SDK\_version
  - IDT\_UniPayI\_V, 101
- sharedController
  - APDUResponse, 77
  - IDT\_UniPayI\_V, 101

IDTEMVData, [104](#)  
IDTMSRData, [107](#)  
swipeMSRData:  
    IDT\_UniPayI\_V\_Delegate-p, [103](#)  
TerminalData, [110](#)