



# **IDTech iOS/OSX SDK Guide for BTPay 200**

**#80121503-001**

**Rev. A**



## Revision History

Revision	Description and Reason for Change	Date
A	Initial Release - Manual;User;BTPay200;SDK;iOS	2/23/2015

# Contents

<b>1</b>	<b>IDTech iOS/OSX SDK Reference Guide for BTPay 200</b>	<b>1</b>
<b>2</b>	<b>Important Security Notice</b>	<b>2</b>
2.1	Applicability	2
2.2	What Does PA-DSS Mean to You?	2
2.3	Third Party Applications	3
2.4	PA-DSS Guidelines	3
2.5	More Information	8
<b>3</b>	<b>BTPay 200 Main Transaction Commands</b>	<b>10</b>
3.1	EMV Methods	10
3.2	PINPad Methods	11
3.3	Display	12
3.4	MSR	12
<b>4</b>	<b>Connecting to BTPay 200 iOS</b>	<b>13</b>
4.1	Connect with Bluetooth: iOS	13
4.2	How to check Bluetooth status	13
4.3	Unpairing BTPay from an iOS device	13
4.4	Troubleshooting Bluetooth connections	14
4.5	Bluetooth disconnects, intermittent connectivity, or unable to find	14
4.6	Can't pair to BTPay	14
4.7	Can't connect with BTPay	15
<b>5</b>	<b>Connecting to BTPay 200 OSX</b>	<b>16</b>
5.1	Connect with USB-HID	16
5.2	Connect with Bluetooth	16
5.3	How to check Bluetooth status	16
5.4	Unpairing BTPay from a Mac	16
5.5	Troubleshooting Bluetooth connections	17
5.6	Bluetooth disconnects, intermittent connectivity, or unable to find	17
5.7	Can't pair to BTPay	17
5.8	Can't connect with BTPay	17

<b>6</b>	<b>Core Implementation BTPay 200: iOS</b>	<b>18</b>
6.1	Integrating with IDTech framework	18
6.2	Import the necessary framework/libraries	18
6.3	Add Import statements to utilize frameworks	23
6.4	Amend the view controller interface to include the framework delegate classes:	23
6.5	Implement any/all of the optional delegate protocols used to receive data from IDT_BTPay_↔ Delegate:	23
6.6	Call the Singleton instance of the IDT_BTPay framework object:	24
6.7	Add the External Accessory device protocol strings to your projects .plist:	24
6.8	Sample Project Tutorial	24
6.8.1	Step 1: Create New Project	25
6.8.2	Step 2: Import Frameworks	26
6.8.3	Step 3: Add Protocol Strings to .plist	26
6.8.4	Step 4: Design Interface	26
6.8.5	Step 5: Configure Header File	27
6.8.6	Step 6: Configure Method File	29
<b>7</b>	<b>Core Implementation IDTech Devices: OSX</b>	<b>32</b>
7.1	Integrating with IDTech framework	32
7.2	Import the necessary framework/libraries	32
7.3	Add Import statements to utilize frameworks	36
7.4	Amend the view controller interface to include the framework delegate classes:	36
7.5	Implement any/all of the optional delegate protocols used to receive data from IDTech framework:	37
7.6	Call the Singleton instance of the IDT_BTPay framework object:	37
7.7	Sample Project Tutorial	38
7.7.1	Step 1: Create New Project	39
7.7.2	Step 2: Import Frameworks	40
7.7.3	Step 3: Design Interface	40
7.7.4	Step 4: Configure Header File	41
<b>8</b>	<b>BTPay Error Code Reference</b>	<b>56</b>
<b>9</b>	<b>Enumeration Reference</b>	<b>58</b>
<b>10</b>	<b>EMV Tag Reference</b>	<b>61</b>
<b>11</b>	<b>General Message Table</b>	<b>70</b>
<b>12</b>	<b>Hierarchical Index</b>	<b>74</b>
12.1	Class Hierarchy	74
<b>13</b>	<b>Data Structure Index</b>	<b>75</b>
13.1	Data Structures	75

<b>14 Data Structure Documentation</b>	<b>76</b>
14.1 AIDEntry Struct Reference	76
14.1.1 Detailed Description	76
14.2 APDUDData 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 APDUResponse Class Reference	77
14.3.1 Detailed Description	78
14.3.2 Method Documentation	78
14.3.2.1 clear	78
14.3.2.2 sharedController	78
14.4 ApplicationID Struct Reference	78
14.4.1 Detailed Description	80
14.5 CAKey Struct Reference	80
14.5.1 Detailed Description	80
14.6 CRLEntry Struct Reference	81
14.6.1 Detailed Description	81
14.7 CTLSResponse Class Reference	81
14.7.1 Detailed Description	82
14.7.2 Method Documentation	82
14.7.2.1 clear	82
14.7.2.2 sharedController	82
14.8 ICCReaderStatus Struct Reference	82
14.8.1 Detailed Description	83
14.9 IDT_BTPay Class Reference	83
14.9.1 Detailed Description	85
14.9.2 Method Documentation	85
14.9.2.1 attemptConnect	85
14.9.2.2 close	85
14.9.2.3 config_getDateTime:	85
14.9.2.4 config_getInterfaceDeviceSN:	85
14.9.2.5 config_getLanguageType:	85
14.9.2.6 config_getModelNumber:	87
14.9.2.7 config_getSerialNumber:	87
14.9.2.8 config_getTerminalIdentification:	87
14.9.2.9 config_setBluetoothAddress:	87
14.9.2.10 config_setCmdTimeOutDuration:	88
14.9.2.11 config_setDateTime:	88

14.9.2.12 config_setInterfaceDeviceSN:	88
14.9.2.13 config_setLanguageType:	88
14.9.2.14 config_setSerialNumber:	89
14.9.2.15 config_setTerminalIdentification:	89
14.9.2.16 config_stayConnected:	89
14.9.2.17 connectUSB:	89
14.9.2.18 device_getFirmwareVersion:	90
14.9.2.19 device_getMaskAndEncryption:	90
14.9.2.20 device_getResponseCodeString:	90
14.9.2.21 device_isConnected:	93
14.9.2.22 device_rebootDevice	93
14.9.2.23 device_restoreMaskAndEncryptionDefaults	93
14.9.2.24 device_sendBeep:numberOfTones:	93
14.9.2.25 device_sendDataCommand:calcLRC:response:	94
14.9.2.26 device_setExpMasking:	94
14.9.2.27 device_setPANMaskingCharacter:	94
14.9.2.28 device_setPostPANClearDigits:	95
14.9.2.29 device_setPrePANClearDigits:	95
14.9.2.30 device_uploadJPEG:	95
14.9.2.31 emv_completeOnlineEMVTransaction:hostResponseTags:responseTags:	96
14.9.2.32 emv_getAllTags:	97
14.9.2.33 emv_getEMVKernelVersion:	97
14.9.2.34 emv_getTag:tagData:	97
14.9.2.35 emv_removeApplicationData:	97
14.9.2.36 emv_removeCAPK:index:	98
14.9.2.37 emv_removeCRL:	98
14.9.2.38 emv_removeCRLUnit:	98
14.9.2.39 emv_removeTerminalData	98
14.9.2.40 emv_retrieveAIDList:	99
14.9.2.41 emv_retrieveApplicationData:response:	99
14.9.2.42 emv_retrieveCAPK:index:response:	99
14.9.2.43 emv_retrieveCAPKList:	100
14.9.2.44 emv_retrieveCRLForRID:response:	100
14.9.2.45 emv_retrieveCRLList:	100
14.9.2.46 emv_retrieveTerminalData:	100
14.9.2.47 emv_setApplicationData:	102
14.9.2.48 emv_setCAPK:	102
14.9.2.49 emv_setCRL:	102
14.9.2.50 emv_setTerminalData:	102

14.9.2.51 emv_startEMVTransaction:otherAmount:timeout:transactionType:additional← Tags:	103
14.9.2.52 icc_exchangeAPDU:response:	104
14.9.2.53 icc_exchangeEncryptedAPDU:ksn:response:	104
14.9.2.54 icc_getAPDU_KSN:	104
14.9.2.55 icc_getICCReaderStatus:	104
14.9.2.56 icc_powerOffICC:	105
14.9.2.57 icc_powerOnICC:	105
14.9.2.58 icc_powerOnICC:response:	105
14.9.2.59 isConnected	106
14.9.2.60 lcd_displayMessage:line2:line3:line4:	106
14.9.2.61 lcd_enterSleepMode	106
14.9.2.62 lcd_setIdleTimeForSleep:	106
14.9.2.63 lcd_showJPEG:Y0:X1:Y1:	107
14.9.2.64 msr_cancelMSRSwipe	107
14.9.2.65 msr_getCardDataEncryptedAlgorithm:	107
14.9.2.66 msr_getEncryptMSRFormat:	107
14.9.2.67 msr_setCardDataEncryptedAlgorithm:	109
14.9.2.68 msr_setEncryptMSRFormat:	109
14.9.2.69 msr_startMSRSwipeWithDisplay:line2:line3:	109
14.9.2.70 pin_cancelPin	110
14.9.2.71 pin_getAmount:maxLength:messageID:language:	110
14.9.2.72 pin_getCardAccount:max:line1:line2:	113
14.9.2.73 pin_getEncryptedData:minLength:maxLength:messageID:language:	113
14.9.2.74 pin_getEncryptedPIN:keyType:line1:line2:line3:	117
14.9.2.75 pin_getFunctionKey	117
14.9.2.76 pin_getNumeric:minLength:maxLength:messageID:language:	119
14.9.2.77 pin_getPinLength:	122
14.9.2.78 pin_getPINpadStatus:	122
14.9.2.79 pin_setPinLength:maxLength:	123
14.9.2.80 SDK_version	123
14.9.2.81 sharedController	123
14.9.3 Property Documentation	123
14.9.3.1 delegate	123
14.10<IDT_BTPay_Delegate> Protocol Reference	124
14.10.1 Detailed Description	124
14.10.2 Method Documentation	124
14.10.2.1 dataInOutMonitor:incoming:	124
14.10.2.2 emvTransactionData:errorCode:	124
14.10.2.3 pinpadData:keySN:event:	125

14.10.2.4 swipeMSRData:	125
14.11 IDT_UniMag Class Reference	125
14.11.1 Method Documentation	126
14.11.1.1 config_getSerialNumber:	126
14.11.1.2 device_connectToAudioReader	126
14.11.1.3 device_getResponseCodeString:	126
14.11.1.4 device_sendUniMagCommand:	127
14.11.1.5 device_setPrePANClearDigits:	127
14.11.1.6 isConnected	127
14.11.1.7 msr_cancelMSRSwipe	127
14.11.1.8 msr_getNextKSN:	128
14.11.1.9 msr_setSwipeEncryption:	129
14.11.1.10 msr_setSwipeForcedEncryptionOption:	129
14.11.1.11 msr_startMSRSwipe	129
14.11.1.12 SDK_version	129
14.11.1.13 sharedController	130
14.11.2 Property Documentation	130
14.11.2.1 delegate	130
14.12 <IDT_UniMag_Delegate> Protocol Reference	130
14.12.1 Detailed Description	130
14.12.2 Method Documentation	131
14.12.2.1 dataInOutMonitor:incoming:	131
14.12.2.2 deviceMessage:	131
14.12.2.3 plugStatusChange:	131
14.12.2.4 swipeMSRData:	131
14.13 IDTCommon Class Reference	131
14.13.1 Detailed Description	132
14.14 IDTech_ApplicationID Struct Reference	132
14.14.1 Detailed Description	132
14.15 IDTech_TerminalData Struct Reference	133
14.15.1 Detailed Description	133
14.16 IDTechEMV Class Reference	133
14.16.1 Detailed Description	134
14.16.2 Method Documentation	134
14.16.2.1 accelerateRead:	134
14.16.2.2 cancelTransaction	134
14.16.2.3 completeOnlineEMVTransaction:resultCode:issuerAuthenticationData:issuerScripts:	135
14.16.2.4 confirmApplication:	135
14.16.2.5 confirmApplicationCancel	135



14.16.2.6	getAllTags	135
14.16.2.7	getEMVKernelVersion	135
14.16.2.8	getTag:	136
14.16.2.9	preloadEMVCardData	136
14.16.2.10	removeApplicationData:	136
14.16.2.11	retrieveAIDList	136
14.16.2.12	retrieveApplicationData:	136
14.16.2.13	retrieveTerminalData	137
14.16.2.14	sendSystemError	137
14.16.2.15	setApplicationData:	137
14.16.2.16	setTag:tagData:	137
14.16.2.17	setTerminalData:	137
14.16.2.18	sharedController	138
14.16.2.19	startEMVTransaction:timeout:transactionType:additionalTags:	138
14.16.3	Property Documentation	138
14.16.3.1	delegate	139
14.17	<IDTechEMV_Delegate> Protocol Reference	139
14.17.1	Detailed Description	139
14.17.2	Method Documentation	139
14.17.2.1	confirmApplicationSelection:retry:	139
14.17.2.2	emvTransactionData:errorCode:performReversal:	139
14.17.2.3	emvTransactionMessage:	140
14.17.2.4	languagePreference:	140
14.18	IDTEMVData Class Reference	140
14.18.1	Detailed Description	141
14.18.2	Method Documentation	141
14.18.2.1	clear	141
14.18.2.2	sharedController	141
14.18.3	Property Documentation	141
14.18.3.1	resultCode	141
14.19	IDTMSRData Class Reference	142
14.19.1	Detailed Description	143
14.19.2	Method Documentation	143
14.19.2.1	clear	143
14.19.2.2	sharedController	143
14.19.3	Property Documentation	143
14.19.3.1	captureEncodeType	143
14.19.3.2	captureEncryptType	144
14.19.3.3	event	144
14.19.3.4	readStatus	144

---

14.20MaskAndEncryption Struct Reference . . . . .	144
14.20.1 Detailed Description . . . . .	145
14.20.2 Field Documentation . . . . .	145
14.20.2.1 encryptionOption . . . . .	145
14.20.2.2 maskOption . . . . .	145
14.21PowerOnStructure Struct Reference . . . . .	145
14.21.1 Detailed Description . . . . .	146
14.22TerminalData Struct Reference . . . . .	146
14.22.1 Detailed Description . . . . .	147
14.23UncaughtExceptionHandler Class Reference . . . . .	147

## Chapter 1

# IDTech iOS/OSX SDK Reference Guide for BTPay 200



IDTech.framework is an Apple Framework that will be provided by IDTech as the main interface between iOS and OSX applications, the BTPay 200 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 or OSX applications wishing to deploy with the payment application.

- [Connecting To BTPay 200 iOS](#)
- [Connecting To BTPay 200 OSX](#)
- [Core Implementation BTPay 200: iOS](#)
- [Core Implementation BTPay 200: OSX](#)
- [General Message Table](#)
- [Important Security Notice](#)
- [BTPay 200 Main Transaction Commands](#)
- [EMV Tag Reference](#)
- [Enumeration Reference](#)
- [BTPay Error Code Reference](#)

## Chapter 2

# Important Security Notice

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

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

### 2.1 Applicability

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

### 2.2 What Does PA-DSS Mean to You?

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

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

# BTPay 200 Main Transaction Commands

The methods below are provided as a reference to the main commands needed to execute an EMV transaction, collect PADPad input, or perform a swipe. Some commands not applicable on some devices.

### 3.1 EMV Methods

#### Start EMV Transaction

`emv_startEMVTransaction:otherAmount:timeout:transactionType:additionalTags: (IDT_BTPay)`

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

#### Complete Online EMV Transaction

`emv_completeOnlineEMVTransaction:hostResponseTags:responseTags: (IDT_BTPay)`

After receiving a host response, pass host tags (minimum 8A Authorization Response Code) as a dictionary 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_BTPay)`  
`emv_removeTerminalData (IDT_BTPay)`  
`emv_setTerminalData: (IDT_BTPay)`

Methods for terminal configuration. When setting the terminal data, you populate and pass and TerminalFile structure. When retrieving terminal data, you can receive the results in the TerminalFile structure in IDTResult.TerminalData

#### AID Management

`emv_retrieveApplicationData:response: (IDT_BTPay)`  
`emv_removeApplicationData: (IDT_BTPay)`  
`emv_setApplicationData: (IDT_BTPay)`  
`emv_retrieveAIDList: (IDT_BTPay)`

Methods for AID management. When setting the AID, you populate and pass and AID structure. When retrieving AID, you can receive the results in the AID structure in IDTResult.aidResponse. When retrieving the AID list, the list of AID Names/length can be retrieved from the populated NSArray

**CAPK Management**

```

emv_retrieveCAPK:index:response: (IDT_BTPay)
emv_removeCAPK:index: (IDT_BTPay)
emv_setCAPK: (IDT_BTPay)
emv_retrieveCAPKList: (IDT_BTPay)

```

Methods for Certificate Authority Public Key management. When setting the CAPK, you populate and pass and [CAKey](#) structure. When specifying a CAPK to retrieve or remove, you populate the relevant fields in the [CAKey](#) structure. When retrieving CAPK, you can receive the results in the [CAKey](#) structure in IDTResult.caKeyData. When retrieving the CAPK list, the list of RID/Index can be retrieved from IDTResult.arrayResult with an array of NSData items, 6 bytes each, bytes 1-5 RID, byte 6 index

**CRL Management**

```

emv_retrieveCRLForRID:response: (IDT_BTPay)
emv_removeCRL: (IDT_BTPay)
emv_removeCRLUnit: (IDT_BTPay)
emv_setCRL: (IDT_BTPay)

```

Methods for Certificate Revocation List management. When setting the CRL, you populate and pass and [CRL↔Entry](#) structure. When retrieving CRL data, reference IDTResult.arrayResult for either an NSArray of NSData 5-byte objects for each RID,, or a NSArray of NSData objects for each serial number found. When specifying CRL, you use [CRL↔Entry](#) structure

**Kernel Version**

```

emv_getEMVKernelVersion: (IDT_BTPay)

```

Method to retrieve kernel version. Valued returned in IDTResult.data

**APDU Communication**

```

icc_exchangeAPDU:response: (IDT_BTPay)
icc_exchangeEncryptedAPDU:ksn:response: (IDT_BTPay)

```

Allows the direct sending of APDU packets to ICC

## 3.2 PINPad Methods

**Encrypted Text Message Parameter**

```

pin_getAmount:maxLength:messageID:language: (IDT_BTPay)
pin_getNumeric:minLength:maxLength:messageID:language: (IDT_BTPay)
pin_getEncryptedData:minLength:maxLength:messageID:language: (IDT_BTPay)

```

These methods prompt for input on the PINpad. They are restricted on the types of messages they can display. A list of approved messages is provided in the [General Message Table](#), or messages can be encoded and provided through request to IDTech. Pinpad input results are returned to the delegate protocol:

```

- (void) pinpadData:(NSData*)encryptedData entry:(NSString*)inputValue pinBlock:(NSData*)pinblock KeySN:(
    NSData*)KSN event:(EVENT_PINPAD_Types)event;

```

**Clear Text Message Parameter**

```

pin_getCardAccount:max:line1:line2: (IDT_BTPay) pin_getEncryptedPIN:keyType:line1:line2:line3: (IDT_BT↔
    Pay)

```

These methods prompt for input on the PINpad. Pinpad input results are returned to the delegate protocol:

```
- (void) pinpadData:(NSData*)encryptedData entry:(NSString*)inputValue pinBlock:(NSData*)pinblock KeySN:(NSData*)KSN event:(EVENT_PINPAD_Types)event;
```

### 3.3 Display

[lcd\\_displayMessage:line2:line3:line4: \(IDT\\_BTPay\)](#)

Displays up to 4 lines of text

[lcd\\_showJPEG:Y0:X1:Y1: \(IDT\\_BTPay\)](#)

Displays a picture uploaded with [device\\_uploadJPEG: \(IDT\\_BTPay\)](#)

### 3.4 MSR

#### Encryption Type

[msr\\_setEncryptMSRFormat: \(IDT\\_BTPay\)](#) [IDT\\_BTPay::msr\\_getEncryptMSRFormat\(\)](#)

Sets and gets the encrypted MSR Data Output Format. Can be encrypted with Data Key or PIN key.

#### Request Swipe

[msr\\_startMSRSwipeWithDisplay:line2:line3: \(IDT\\_BTPay\)](#)

Enables MSR to receive Swipe. Results are returned as [IDTMSRData](#) in [swipeMSRData](#)

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

#### Cancel Swipe

[msr\\_cancelMSRSwipe \(IDT\\_BTPay\)](#)

Disables the MSR from receiving swipes.

## Chapter 4

# Connecting to BTPay 200 iOS

The BTPay connects through Bluetooth on IOS.

### 4.1 Connect with Bluetooth: iOS

You can use your iPhone, iPad and iPod touch the BTPay. Before you can use the BTPay with an iOS device, you need to pair it

- Make sure your BTPay is on and displaying the default power-on screen
- From the Home screen on your iOS, choose Settings > Bluetooth, the iOS device searches for nearby BTPay.

Note: While your iOS device can maintain multiple pairing records, it can only connect to one BTPay at a time. This prevents your iOS device from sending your data to the wrong BTPay.

- Choose the BTPay, and then enter a PIN "123456".

When pairing is complete, you can use the BTPay with your iOS device.

### 4.2 How to check Bluetooth status

You can see whether a Bluetooth accessory is connected to an iOS device, by looking at the Bluetooth icon in the status bar at the top of the screen:

- Blue or White: Bluetooth is on and a device is connected to the iOS device.
- Gray: Bluetooth is on but no device is connected. If you've paired a device with an iOS device, it may be out of range or turned off.
- No Bluetooth icon in status bar: Bluetooth has not been paired with any devices.

### 4.3 Unpairing BTPay from an iOS device

If you've paired your iOS device with BTPay and no longer wish to use that accessory, you may unpair it. If you wish to leave the device paired, but route calls through a second, paired device, turn off the first device and then turn on the second device. Then in Settings > Bluetooth, tap on the device you wish to use.

- From the Home screen choose Settings > Bluetooth. If Bluetooth isn't on, turn it on.

- Choose a device and tap Unpair.
- Until you pair the Bluetooth accessory again, your iOS device won't route information through it.

## 4.4 Troubleshooting Bluetooth connections

Occasionally while using your iOS device you may notice unexpected Bluetooth behavior such as disconnects, intermittent connectivity, or difficulty finding, pairing, or connecting to BTPay.

Follow these steps first:

- Check that you are in range of the BTPay with which you are trying to pair your iPhone, iPad, or iPod touch. Most Bluetooth accessories have a range of approximately 30 feet.
- On your iOS device, tap Settings > Bluetooth and find BTPay in the list. If BTPay says Not Connected, tap the name of BTPay to attempt to connect it.
- Tap Settings > Bluetooth and turn Bluetooth off and then on again.
- Make sure that your BTPay is turned on and fully charged or connected to power.
- Restart your iOS device.
- Tap Settings > Bluetooth and locate the Bluetooth accessory you are currently connected to or attempting to connect to. Then tap information icon on the right and "Forget this Device". Try to pair your Bluetooth accessory again. You will enter your BTPay PIN "123456" again.
- Restart the BTPay accessory by unplugging from USB power and pulling battery.

## 4.5 Bluetooth disconnects, intermittent connectivity, or unable to find

- Move your iOS device closer to the BTPay.
- Position your iOS device and BTPay in direct line of sight of each other. For example, move the BTPay and iOS device to the same side of your body.
- Remove any case, stand, or other accessories from your iOS device and see if performance improves.
- Check for sources of potential interference and move both the iOS device and the BTPay away from other electronic devices.
- Reset network settings by tapping Settings > General > Reset > Reset Network Settings. This will reset all network settings including: previously connected Wi-Fi networks and passwords, recently used Bluetooth accessories, VPN, and APN settings.

## 4.6 Can't pair to BTPay

- If you see the word "Connected" next to the BTPay in Settings > Bluetooth, your iOS device is already paired with BTPay. Remember, you can only connect to one BTPay at a time.
- Be sure that you are entering the correct PIN "123456" when pairing with BTPay.
- Reset network settings by tapping Settings > General > Reset > Reset Network Settings. This will reset all network settings including: previously connected Wi-Fi networks and passwords, recently used Bluetooth accessories, VPN and APN settings.



## 4.7 Can't connect with BTPay

- Move your iOS device closer to the BTPay.
- Remove any case, stand, or other accessories from your iOS device and see if performance improves.
- Check for sources of potential interference and move both the iOS device and the BTPay away from other electronic devices.
- Reset network settings by tapping Settings > General > Reset > Reset Network Settings. This will reset all network settings including: previously connected Wi-Fi networks and passwords, recently used Bluetooth accessories, VPN and APN settings.

## Chapter 5

# Connecting to BTPay 200 OSX

The BTPay connects through Bluetooth or USB-HID on OSX.

### 5.1 Connect with USB-HID

The BTPay will be recognized as a Human Interface Device once plugged into the Macintosh USB port. Use a standard USB-miniUSB plug to attach the BTPay (mini-USB port on left side) to the Macintosh.

### 5.2 Connect with Bluetooth

You can use your Macintosh with the BTPay. Before you can use the BTPay with your Mac, you need to pair it

- Make sure your BTPay is on and displaying the default power-on screen
- From the Mac desktop, choose System Preferences > Bluetooth, the mac searches for nearby BTPay.

Note: While your Mac device can maintain multiple pairing records, it can only connect to one BTPay at a time. This prevents your Mac from sending your data to the wrong BTPay.

- Once BTPay is discovered, click the "Pair" button next to BTPay.
- Because a PIN is required, the pairing will initially fail. When it fails, another button will appear "Options...". Click that button.
- Enter the PIN "123456" and Pair. When pairing is complete, you can use the BTPay with your Mac.

### 5.3 How to check Bluetooth status

You can see whether a Bluetooth accessory is connected to the Mac, by looking at the Bluetooth icon in the status bar at the top of the screen:

- Clicking on that icon will present a drop down menu with all paired accessories

### 5.4 Unpairing BTPay from a Mac

If you've paired your Mac with BTPay and no longer wish to use that accessory, you may unpair it. If you wish to leave the device paired, but route calls through a second, paired device, turn off the first device and then turn on the second device. Then in System Preferences > Bluetooth, tap on the device you wish to use.

- From the Home screen choose System Preferences > Bluetooth. If Bluetooth isn't on, turn it on.
- Choose the BTPay and click the close "X" icon on the right of the name.
- Until you pair the Bluetooth accessory again, your Mac won't route information through it.

## 5.5 Troubleshooting Bluetooth connections

Occasionally while using your Mac you may notice unexpected Bluetooth behavior such as disconnects, intermittent connectivity, or difficulty finding, pairing, or connecting to BTPay.

Follow these steps first:

- Check that you are in range of the BTPay with which you are trying to pair your Mac. Most Bluetooth accessories have a range of approximately 30 feet.
- On your Mac, select System Preferences > Bluetooth and find BTPay in the list. If BTPay says Not Connected, tap the name of BTPay to attempt to connect it.
- select System Preferences > Bluetooth and turn Bluetooth off and then on again.
- Make sure that your BTPay is turned on and fully charged or connected to power.
- Restart your Mac.
- Select System Preferences > Bluetooth and locate the Bluetooth accessory you are currently connected to or attempting to connect to. Then tap close icon "X" on the right and remove the device. Try to pair your Bluetooth accessory again. You will to enter your BTPay PIN "123456" again.
- Restart the BTPay accessory by unplugging from USB power and pulling battery.

## 5.6 Bluetooth disconnects, intermittent connectivity, or unable to find

- Move your BTPay closer to your Mac.
- Position your Mac and BTPay in direct line of sight of each other.
- Check for sources of potential interference and move both the Mac and the BTPay away from other electronic devices.

## 5.7 Can't pair to BTPay

- If you see the word "Connected" next to the BTPay in System Preferences > Bluetooth, your Mac is already paired with BTPay. Remember, you can only connect to one BTPay at a time.
- Be sure that you are entering the correct PIN "123456" when pairing with BTPay.

## 5.8 Can't connect with BTPay

- Move your BTPay closer to your Mac.
- Check for sources of potential interference and move both the Mac and the BTPay away from other electronic devices.

## Chapter 6

# Core Implementation BTPay 200: iOS

IDTech Framework includes class libraries to interface with the BTPay 200. 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\\_BTPay objects](#)
- [Add Protocol Strings to .plist](#)
- [Sample Project Tutorial](#)

### 6.2 Import the necessary framework/libraries

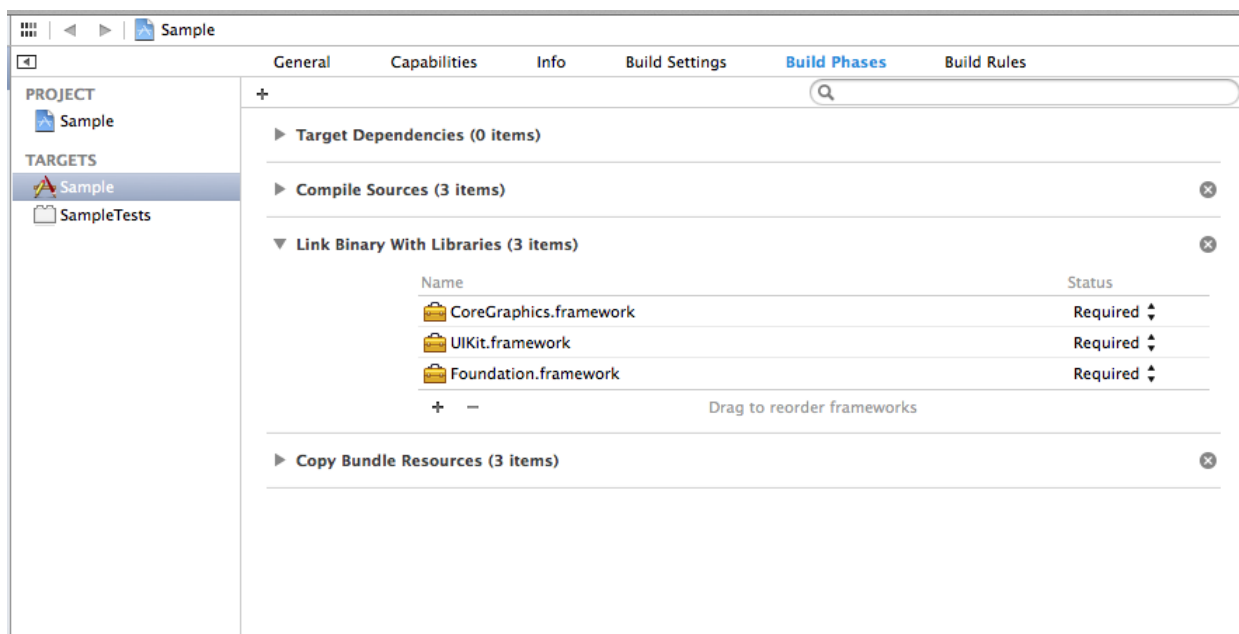
Communicating with IDTech Devices requires the following framework/libraries to be imported into the project:

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

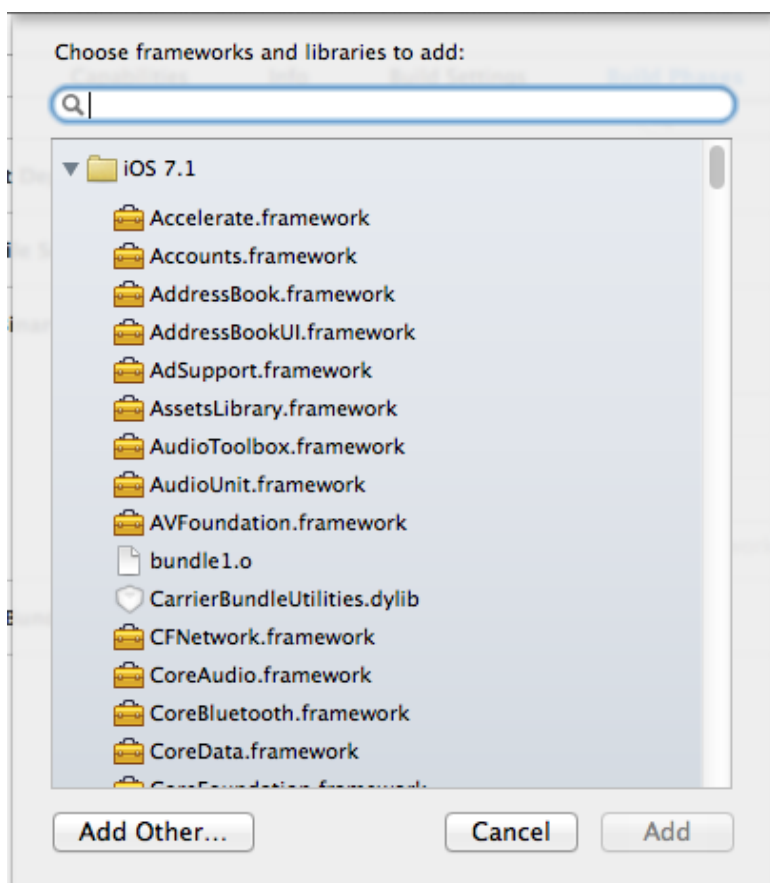
Also, import the following resource bundle:

- IDTech.bundle

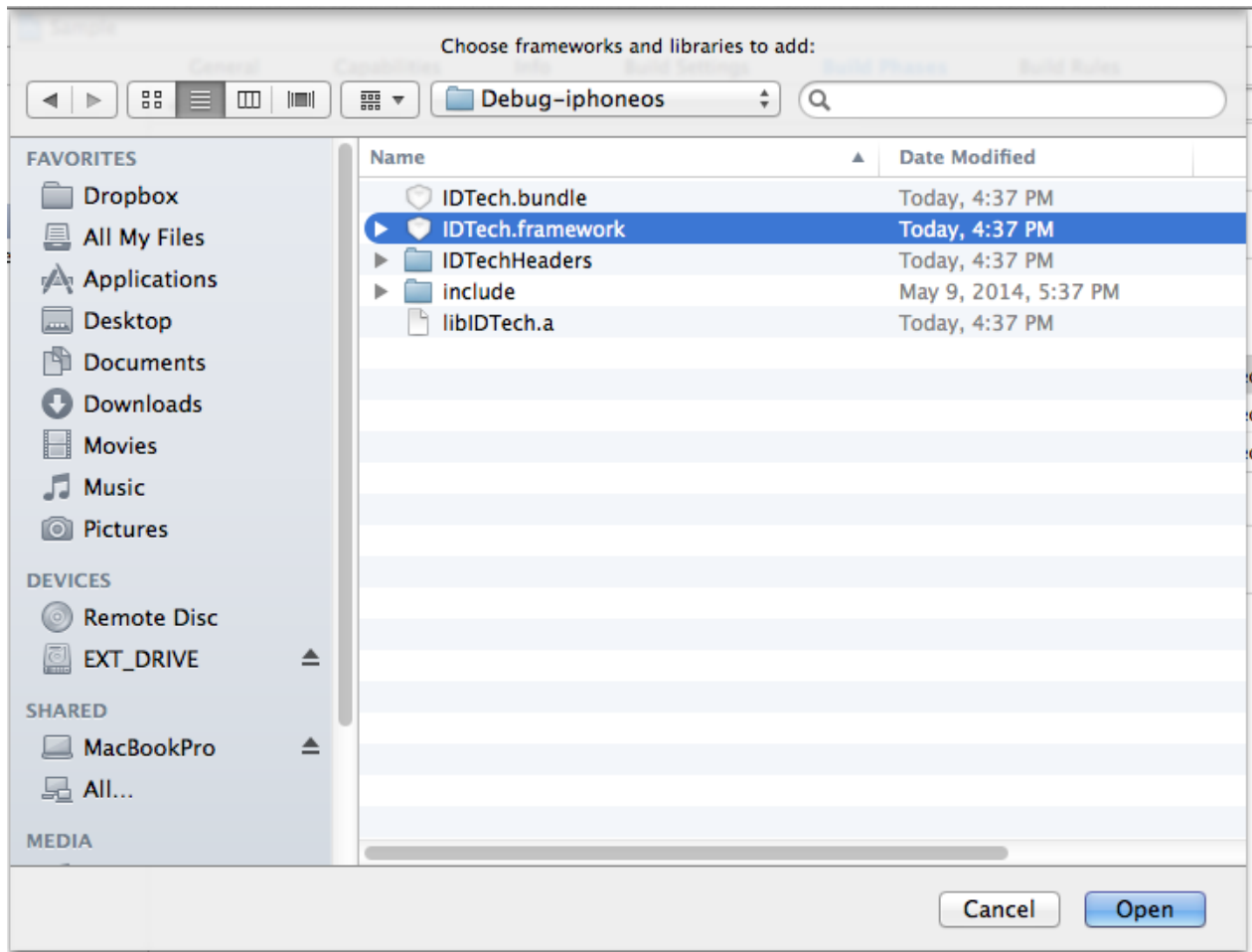
Under Build Phases, select Link Binary With Libraries and click the Add (+) button



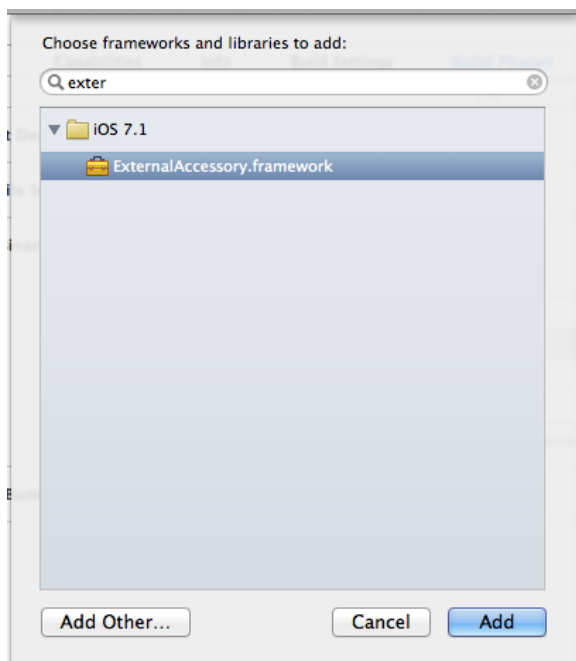
On the Choose Frameworks screen, click "Add Other" in the lower left



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

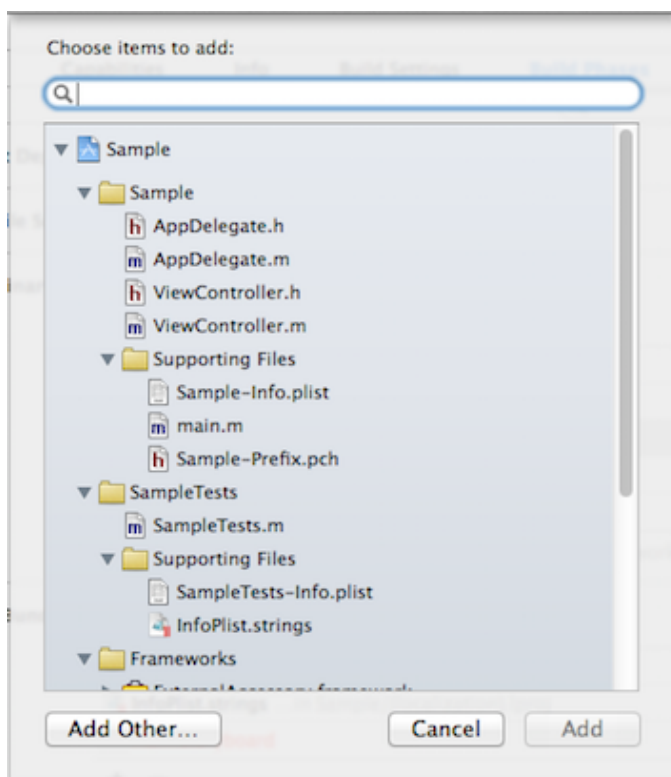
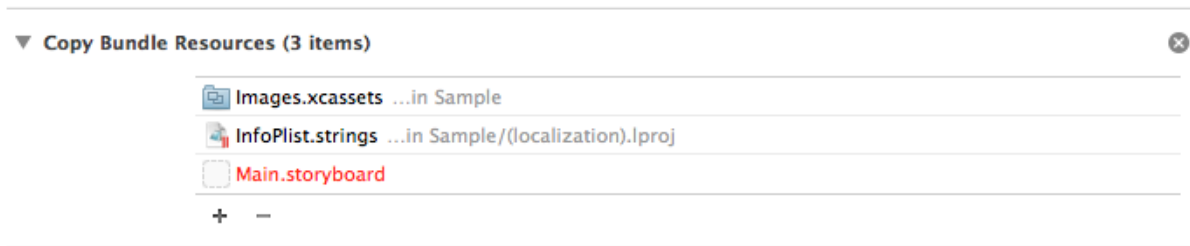


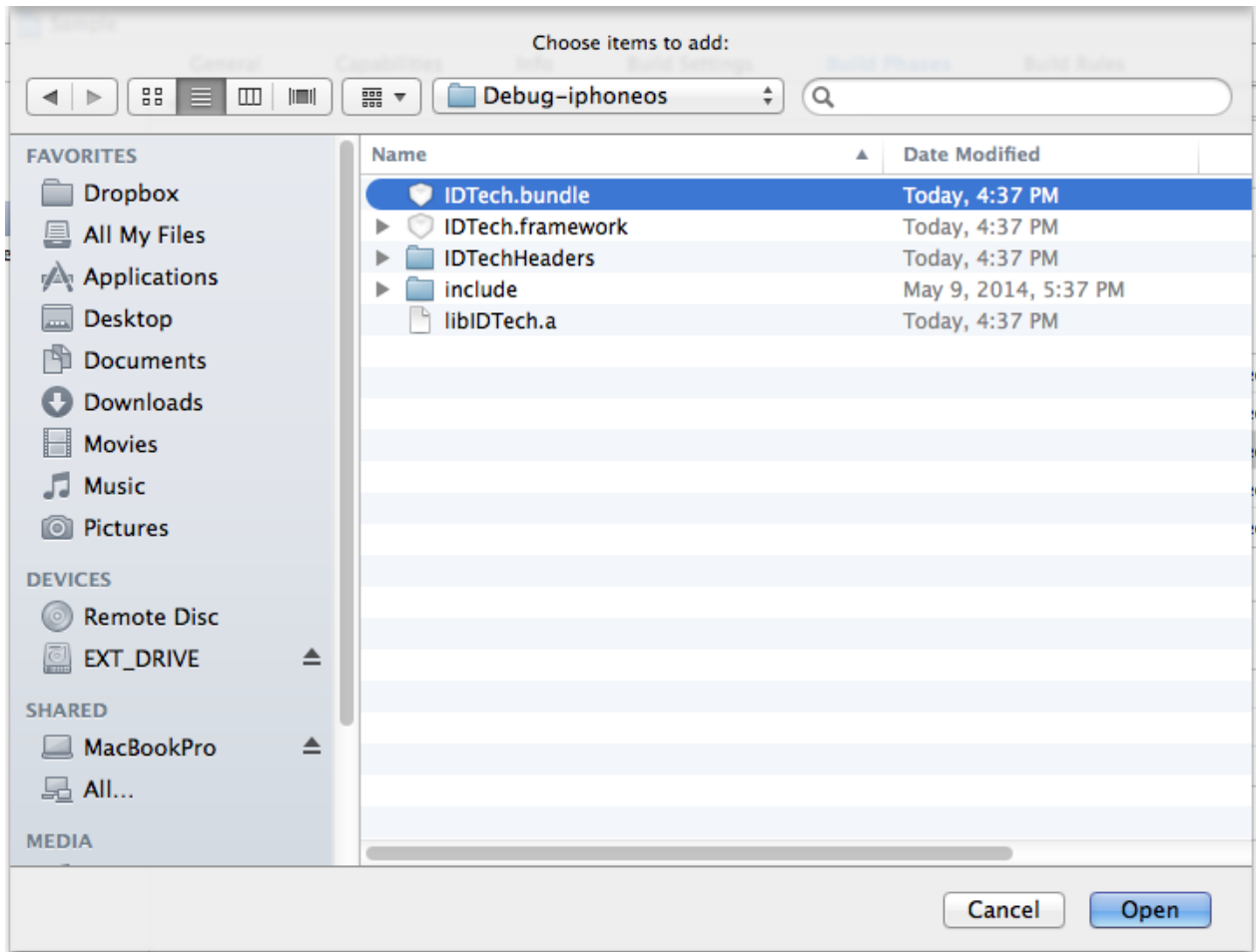
Link the ExternalAccessory framework. On the Choose Frameworks screen, type "exter" into the search bar, select ExternalAccessory.framework and click "Open"



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

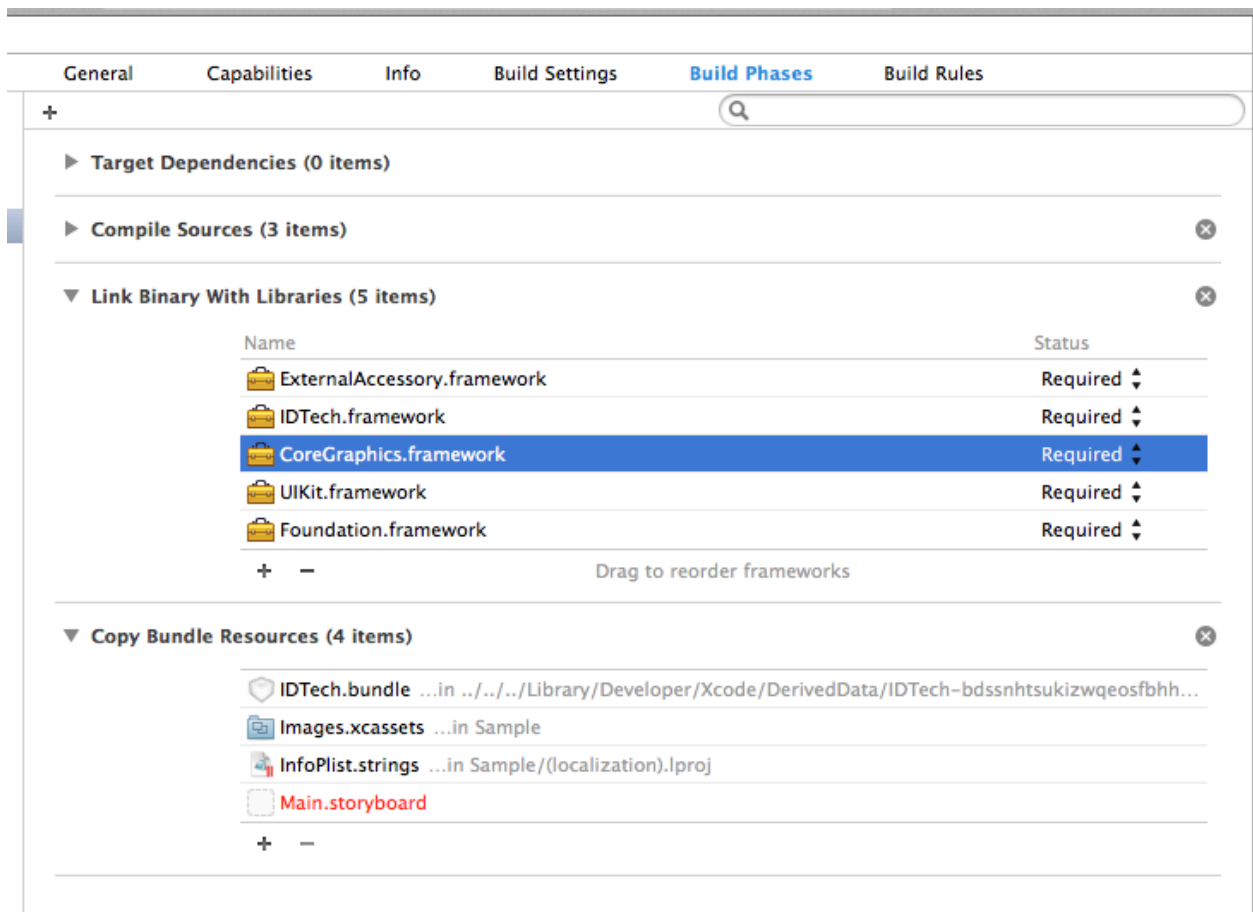
Link another library. Under Copy Bundle, click the Add (+) button, click "Add Other", navigate to and select the IDTech.bundle file and click "Open"





The Build Phases should now include the required frameworks/libraries for the BTPay 200





## 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_BTPay_Delegate>
```

## 6.5 Implement any/all of the optional delegate protocols used to receive data from IDT\_BTPay\_Delegate:

```
-(void) deviceConnected{
}

-(void) deviceDisconnected{
}

-(void) swipeMSRData:(IDTMSRData*) cardData{
}
```

```

- (void) pinpadData:(NSData*)value keySN:(NSData*)KSN event:(EVENT_PINPAD_Types)event{
}

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

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

```

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

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

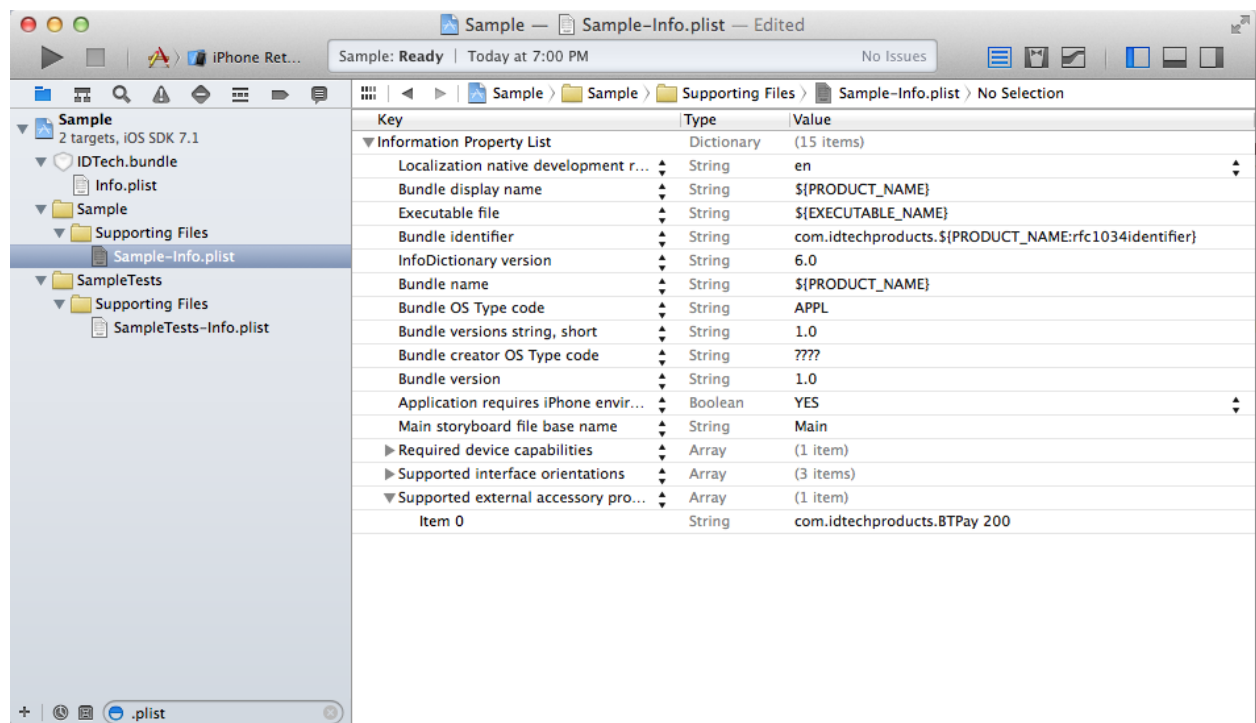
```

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

```

## 6.7 Add the External Accessory device protocol strings to your projects .plist:

When you are using a device that is accessed on iOS over External Accessory (Bluetooth or Dock Connector), the protocol string must be added to your application .plist under "Supported external accessory protocols" array. BTPay 200 protocol string = "com.idtechproducts.BTPay 200":



## 6.8 Sample Project Tutorial

Using Xcode 5.0+, we will create a sample project that will interface with the BTPay200 and will perform the following activities:

- Execute a beep

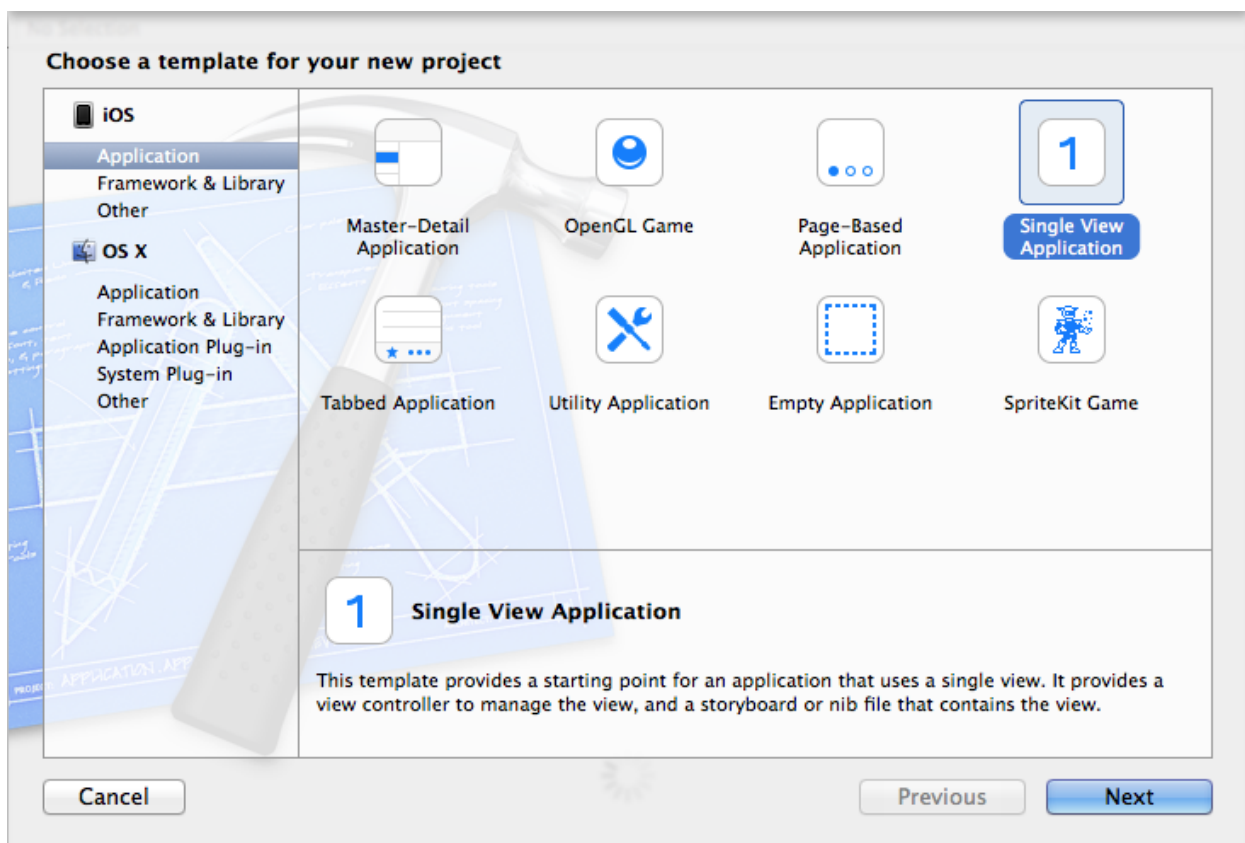
- Display unsecured message
- Turn on/off MSR reader and perform a card capture
- Request a PIN and capture KSN

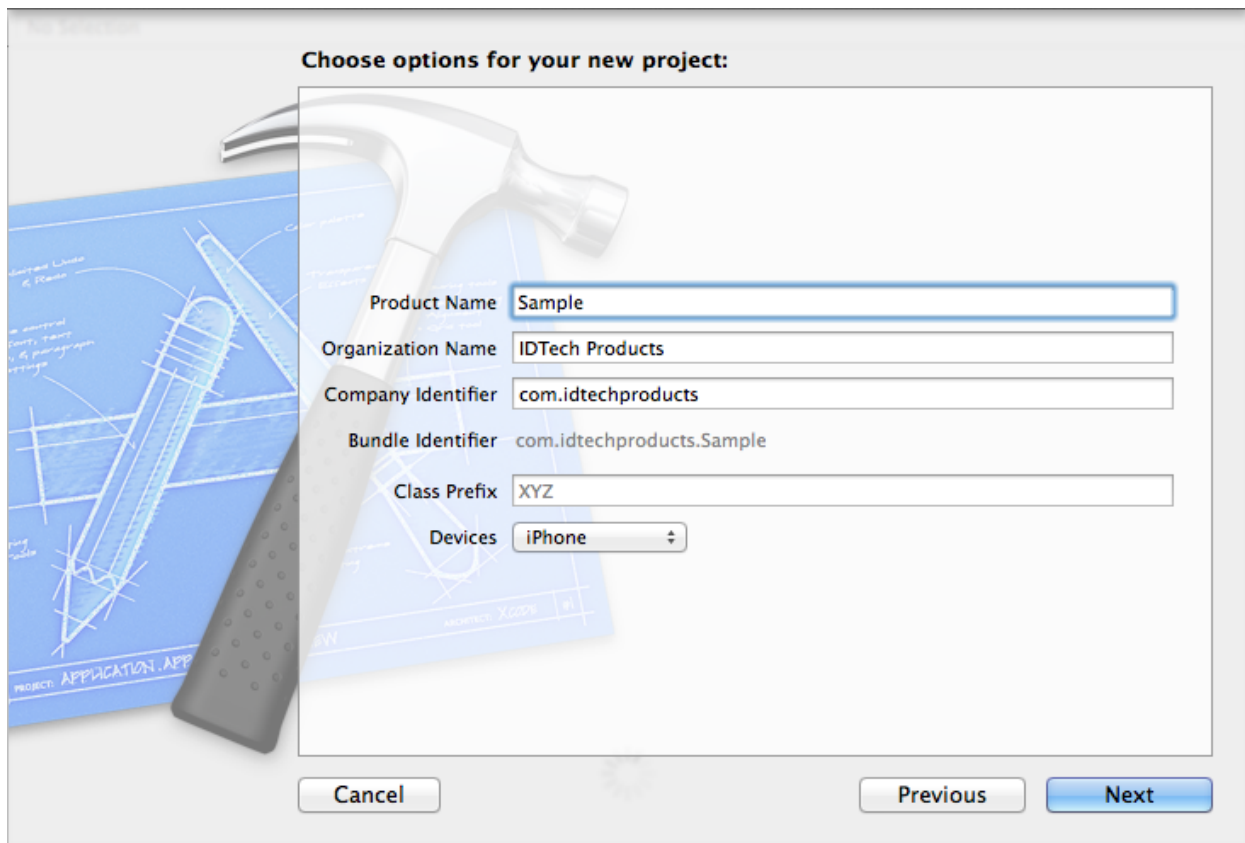
Protocol Delegates:

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

### 6.8.1 Step 1: Create New Project

Create a new Single View Application in Xcode





## 6.8.2 Step 2: Import Frameworks

[Import the necessary framework/libraries](#)

## 6.8.3 Step 3: Add Protocol Strings to .plist

We would like the application to work with apple External Accessory protocol, so we will add the BTPay200 protocol string to the .plist:

[Add Protocol Strings to .plist](#)

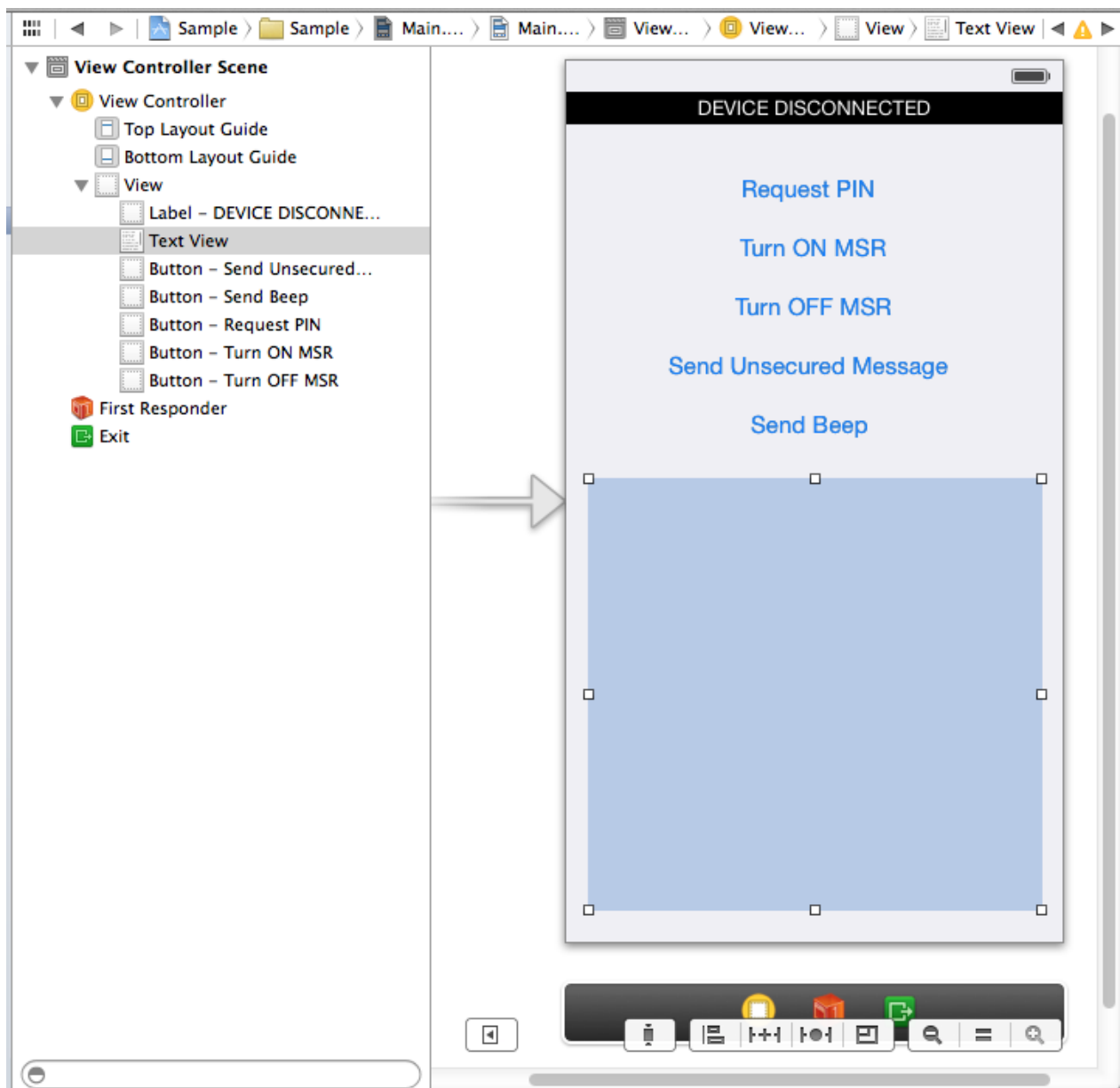
## 6.8.4 Step 4: Design Interface

Design the User Interface by editing the iPhone storyboard file

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

- Add a label to the top that will signify connection/disconnection status.
- Add a text view to communicate data from the BTPay200. 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:
  - Request PIN
  - Turn ON MSR
  - Turn OFF MSR
  - Send Message

– Send Beep



### 6.8.5 Step 5: Configure Header File

In the header file, perform the following:

- [Add Import statements to utilize frameworks](#)
- [Amend the view controller interface](#)
- Create an IBOutlet for the UITextView and link it as a Referencing Outlet to the UITextView on the storyboard
- Create an IBOutlet for the UILabel and link it as a Referencing Outlet to the UILabel on the storyboard
- Create the 5 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>
```

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

@interface ViewController : UIViewController <IDT_BTPay_Delegate>{
    IBOutlet UITextView *tv;
    IBOutlet UILabel *connectedLabel;
}

-(IBAction) msrON:(id)sender;
-(IBAction) msrOff:(id)sender;
-(IBAction) sendUnsecureMessage:(id)sender;
-(IBAction) device_sendBeep:(id)sender;
-(IBAction) requestPIN:(id)sender;

@end
```

## Storyboard Source Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB" version="3.0" toolsVersion="5056"
    systemVersion="13C1021" targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES"
    initialViewController="vXZ-lx-hvc">
    <dependencies>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="3733"/>
    </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="320" height="568"/>
                        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
                        <subviews>
                            <label opaque="NO" clipsSubviews="YES" userInteractionEnabled="NO" contentMode="
                                left" horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES" text="DEVICE
                                DISCONNECTED" textAlignment="center" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines"
                                adjustsFontSizeToFit="NO" translatesAutoresizingMaskIntoConstraints="NO" id="E0F-BN-oKL">
                                <rect key="frame" x="0.0" y="20" width="320" height="21"/>
                                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
                                    YES"/>
                                <color key="backgroundColor" cocoaTouchSystemColor="darkTextColor"/>
                                <fontDescription key="fontDescription" name="HelveticaNeue" family="
                                    Helvetica Neue" pointSize="13"/>
                                <color key="textColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
                                <nil key="highlightedColor"/>
                            </label>
                            <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="
                                scaleToFill" fixedFrame="YES" editable="NO" translatesAutoresizingMaskIntoConstraints="NO" id="B4R-yf-dxd">
                                <rect key="frame" x="14" y="269" width="293" height="279"/>
                                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
                                    YES"/>
                                <color key="backgroundColor" red="1" green="1" blue="1" alpha="1"
                                    colorSpace="calibratedRGB"/>
                                <fontDescription key="fontDescription" type="system" pointSize="14"/>
                                <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
                            </textView>
                            <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
                                contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="
                                middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="uBt-t4-tkZ">
                                <rect key="frame" x="14" y="182" width="286" height="30"/>
                                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
                                    YES"/>
                                <state key="normal" title="Send Unsecured Message">
                                    <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
                                        calibratedWhite"/>
                                </state>
                                <connections>
                                    <action selector="sendUnsecureMessage:" destination="vXZ-lx-hvc"
                                        eventType="touchUpInside" id="TOF-Lp-y60"/>
                                </connections>
                            </button>
                            <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
                                contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="
                                middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="ZqT-dn-dUn">
                                <rect key="frame" x="14" y="220" width="286" height="30"/>
                                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
                                    YES"/>
                                <state key="normal" title="Send Beep">
                                    <color key="titleShadowColor" white="0.5" alpha="1" colorSpace="
                                        calibratedWhite"/>
                                </state>
                                <connections>
                                    <action selector="device_sendBeep:" destination="vXZ-lx-hvc"
                                        eventType="touchUpInside" id="TOF-Lp-y60"/>
                                </connections>
                            </button>
                        </subviews>
                    </view>
                </viewController>
            </objects>
        </scene>
    </scenes>
</document>
```

```

        </state>
        <connections>
            <action selector="device_sendBeep:" destination="vXZ-lx-hvc" eventType="
"touchUpInside" id="Of8-WU-aQJ"/>
        </connections>
    </button>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="
middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="3Xz-6S-Tly">
        <rect key="frame" x="14" y="68" width="286" height="30"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
YES"/>

        <state key="normal" title="Request PIN">
            <color key="titleLabelShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
        </state>
        <connections>
            <action selector="requestPIN:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="eQC-pl-qjT"/>
        </connections>
    </button>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="
middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="841-S2-Myl">
        <rect key="frame" x="17" y="106" width="286" height="30"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
YES"/>

        <state key="normal" title="Turn ON MSR">
            <color key="titleLabelShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
        </state>
        <connections>
            <action selector="msrON:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="jX1-00-N84"/>
        </connections>
    </button>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect" lineBreakMode="
middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="VK1-53-hV1">
        <rect key="frame" x="17" y="144" width="286" height="30"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="
YES"/>

        <state key="normal" title="Turn OFF MSR">
            <color key="titleLabelShadowColor" white="0.5" alpha="1" colorSpace="
calibratedWhite"/>
        </state>
        <connections>
            <action selector="msrOff:" destination="vXZ-lx-hvc" eventType="
touchUpInside" id="hkB-ik-j09"/>
        </connections>
    </button>
</subviews>
<color key="backgroundColor" cocoaTouchSystemColor="groupTableViewBackgroundColor"/>
</view>
<connections>
    <outlet property="connectedLabel" destination="E0F-BN-oKL" id="0Sj-NW-OEl"/>
    <outlet property="tv" destination="B4R-yf-dxd" id="Dxt-QI-I57"/>
</connections>
</viewController>
<placeholder placeholderIdentifier="IBFirstResponder" id="x5A-6p-PRh" sceneMemberID="
firstResponder"/>
</objects>
</scene>
</scenes>
<simulatedMetricsContainer key="defaultSimulatedMetrics">
    <simulatedStatusBarMetrics key="statusBar"/>
    <simulatedOrientationMetrics key="orientation"/>
    <simulatedScreenMetrics key="destination" type="retina4"/>
</simulatedMetricsContainer>
</document>

```

### 6.8.6 Step 6: Configure Method File

In the header file, perform the following:

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

```

- (void)viewDidLoad
{

```

```

[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
[[IDT_BTPay sharedController] setDelegate:self];
}

```

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

```

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

```

- Implement protocol delegate `IDT_BTPayDelegate::pinpadData:keySN:event():` to receive pinpad data. Reference: [Implement optional delegate protocols](#)

```

- (void) pinpadData:(NSData*)value keySN:(NSData*)KSN event:(EVENT_PINPAD_Types)event{
    if (event == EVENT_PINPAD_ENCRYPTED_PIN) {
        tv.text = [NSString stringWithFormat:@"%@\nPINBLOCK: %@",tv.text,value.description];
        tv.text = [NSString stringWithFormat:@"%@\nKSN: %@",tv.text,KSN.description];
    }
}

```

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

```

- (void) swipeMSRData:(IDTMSRData*)cardData{
    switch (cardData.event) {
        case EVENT_MSR_CARD_DATA:
        {
            switch (cardData.encryptionType) {
                case CaptureEncodeType_ISOABA:
                    tv.text = [NSString stringWithFormat:@"%@\nEncryption Type: %@",tv.text, @"ISO/ABA"];
                    break;
                case CaptureEncodeType_AAMVA:
                    tv.text = [NSString stringWithFormat:@"%@\nEncryption Type: %@",tv.text, @"AA/MVA"];
                    break;
                case CaptureEncodeType_Other:
                    tv.text = [NSString stringWithFormat:@"%@\nEncryption Type: %@",tv.text, @"Other"];
                    break;
                case CaptureEncodeType_Undecoded:
                    tv.text = [NSString stringWithFormat:@"%@\nEncryption Type: %@",tv.text, @"Undecoded"];
                    break;
                default:
                    tv.text = [NSString stringWithFormat:@"%@\nEncryption Type: %@",tv.text, @"UNKNOWN"];
                    break;
            }

            tv.text = [NSString stringWithFormat:@"%@\nTrack 1: %@",tv.text, cardData.track1];
            tv.text = [NSString stringWithFormat:@"%@\nTrack 2: %@",tv.text, cardData.track2];
            tv.text = [NSString stringWithFormat:@"%@\nTrack 3: %@",tv.text, cardData.track3];
            tv.text = [NSString stringWithFormat:@"%@\nEncoded Track 1: %@",tv.text, cardData.encTrack1.description];
            tv.text = [NSString stringWithFormat:@"%@\nEncoded Track 2: %@",tv.text, cardData.encTrack2.description];
            tv.text = [NSString stringWithFormat:@"%@\nEncoded Track 3: %@",tv.text, cardData.encTrack3.description];
            tv.text = [NSString stringWithFormat:@"%@\nHash Track 1: %@",tv.text, cardData.hashTrack1.description];
            tv.text = [NSString stringWithFormat:@"%@\nHash Track 2: %@",tv.text, cardData.hashTrack2.description];
            tv.text = [NSString stringWithFormat:@"%@\nHash Track 3: %@",tv.text, cardData.hashTrack3.description];
            tv.text = [NSString stringWithFormat:@"%@\nKSN: %@",tv.text, cardData.KSN.description];

            return;
        }
        break;
        case EVENT_MSR_CANCEL_KEY:
        {

```



```

        tv.text = [NSString stringWithFormat:@"%@\n(Event) MSR Cancel Key received: %@",tv.text,
cardData.encTrack1];
        return;
    }
    break;

    case EVENT_MSR_BACKSPACE_KEY:
    {
        tv.text = [NSString stringWithFormat:@"%@\n(Event) MSR Backspace Key received: %@",tv.text,
cardData.encTrack1];
        return;
    }
    break;

    case EVENT_MSR_ENTER_KEY:
    {
        tv.text = [NSString stringWithFormat:@"%@\n(Event) MSR Enter Key received: %@",tv.text,
cardData.encTrack1];
        return;
    }
    break;

    case EVENT_MSR_UNKNOWN:
    {
        tv.text = [NSString stringWithFormat:@"%@\n(Event) MSR unknown event, data: %@",tv.text,
cardData.encTrack1];
        return;
    }
    break;

    default:
        break;
}
}

```

- Implement the button press methods

```

-(IBAction) msrON:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController] msr_startMSRSwipeWithDisplay:@"Please" line2:@"
swipe card" line3:nil];
    tv.text = [NSString stringWithFormat:@"%@\nEnable MSR Return Status Code %i ",tv.text, rt];
}
-(IBAction) msrOff:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController] msr_cancelMSRSwipe];
    tv.text = [NSString stringWithFormat:@"%@\nDisable MSR Return Status Code %i ",tv.text, rt];
}
-(IBAction) sendUnsecureMessage:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController] lcd_displayMessage:@"This is" line2:@"an
unsecured" line3:@"message." line4:@"Thank you!"];
    tv.text = [NSString stringWithFormat:@"%@\nDisplay Message Return Status Code %i ",tv.text, rt];
}

-(IBAction) device_sendBeep:(id)sender{
    unsigned short beep[] = {0xb00,0x400,0x800,0x300};
    IDT_STATUS rt = [[IDT_BTPay sharedController] device_sendBeep:beep numberOfTones:2];
    tv.text = [NSString stringWithFormat:@"%@\nControl Beep Return Status Code %i ",tv.text, rt];
}
-(IBAction) requestPIN:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController] pin_getEncryptedPIN:@"1234567890123456" keyType:
PIN_KEY_TDES_DUKPT_extp line1:@"Please" line2:@"input PIN" line3:nil];
    tv.text = [NSString stringWithFormat:@"%@\nRequest PIN Return Status Code %i ",tv.text, rt];
}

```

## Chapter 7

# Core Implementation IDTech Devices: OSX

IDTech Framework includes class libraries to interface with the BTPay 200 over Bluetooth and/or USB-HID. This guide assume a fair understanding of Xcode 5.0+ and general Apple OSX programming knowledge.

### 7.1 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\\_BTPay objects](#)
- [Sample Project Tutorial](#)

### 7.2 Import the necessary framework/libraries

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

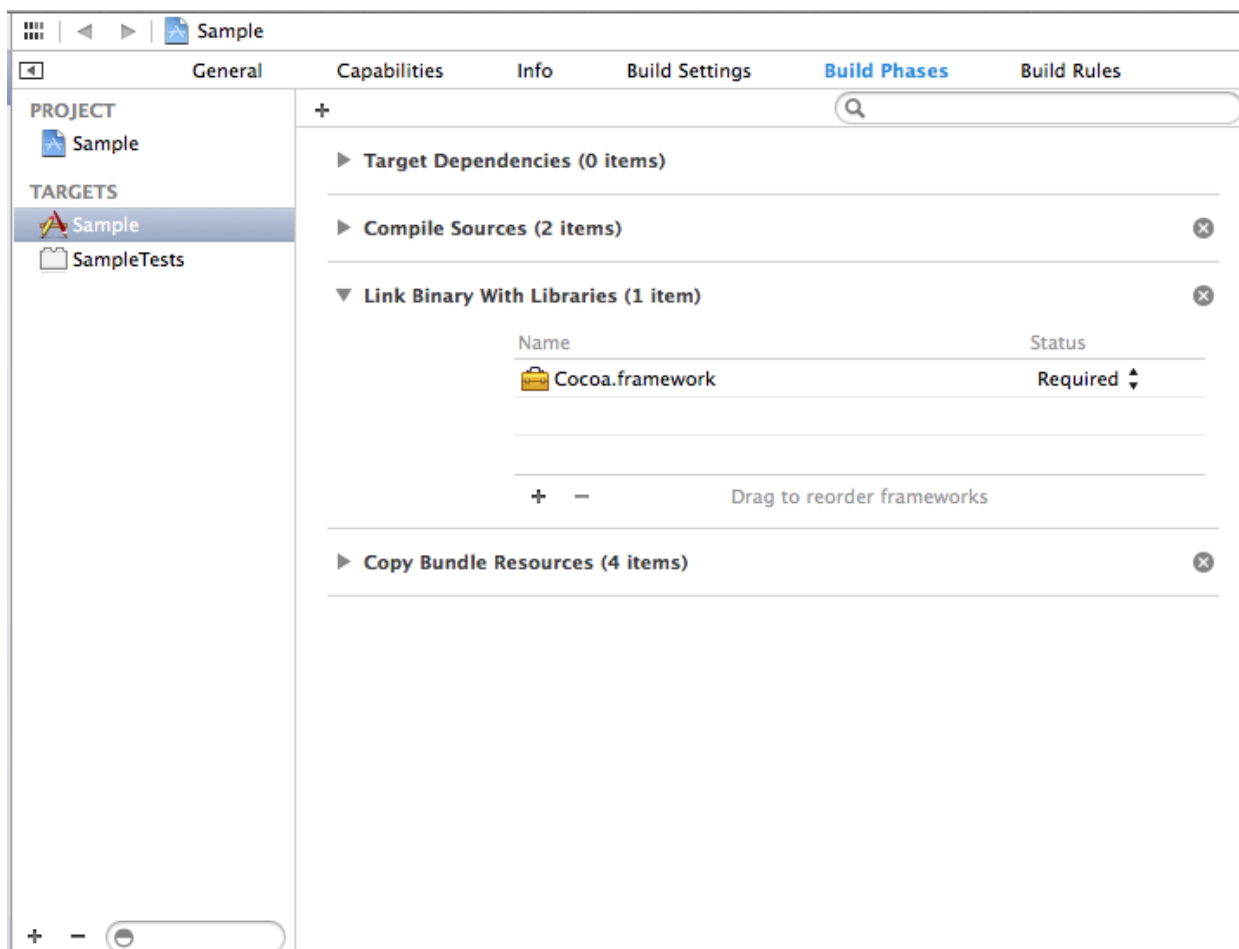
- IDTech.framework

Also, import the following resource bundle:

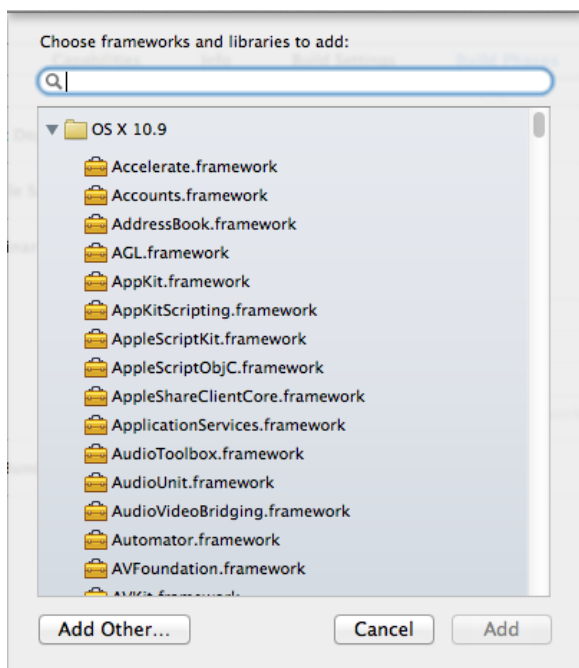
- IDTech.bundle

During runtime, any third party frameworks your cocoa application references must be located in the /Library/Frameworks folder. As a first step, place the IDTech.framework package in the /Library/Frameworks folder. Optionally, you can create a symbolic link to the IDTech.framework within the /Library/Frameworks folder.

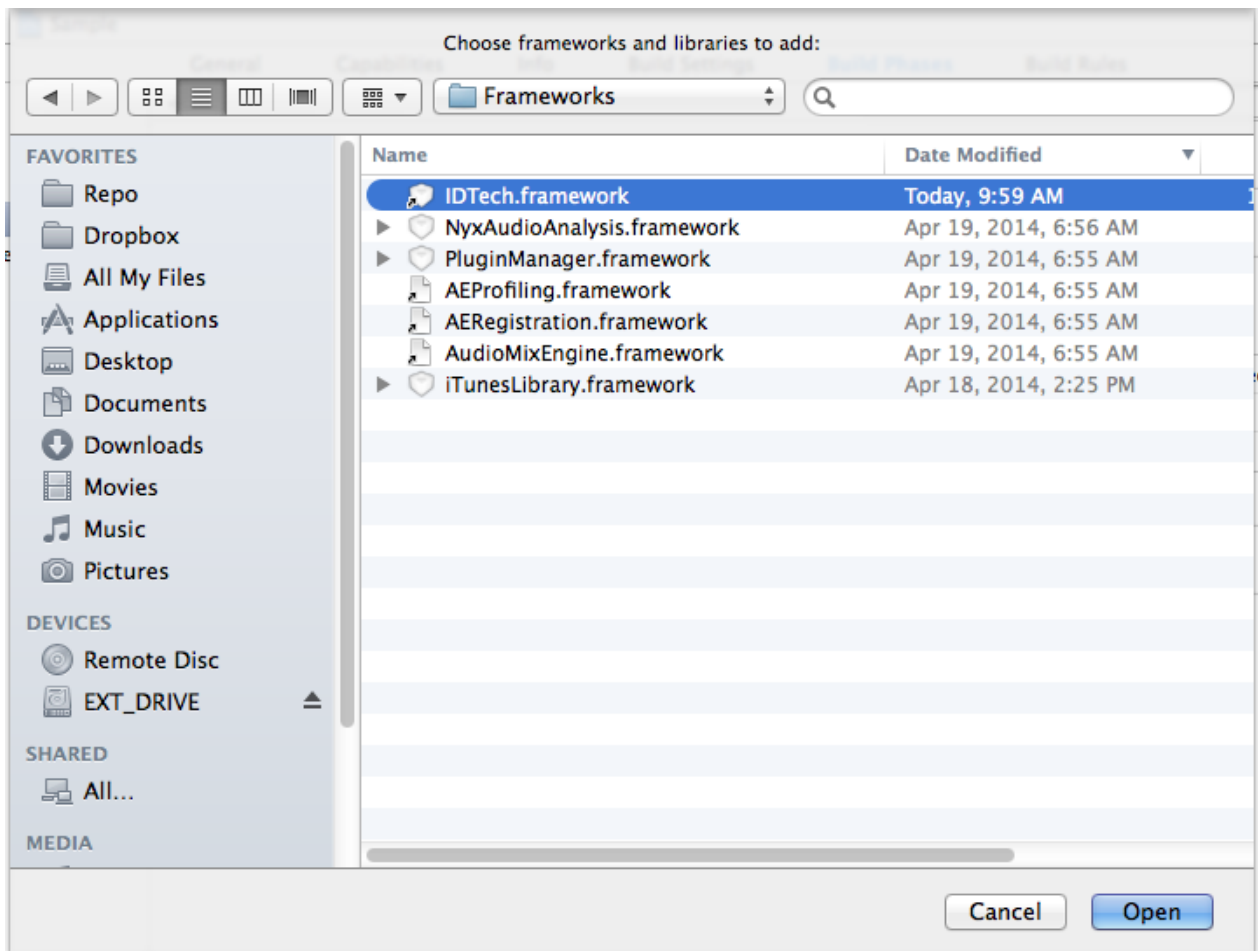
Under Build Phases, select Link Binary With Libraries and click the Add (+) button



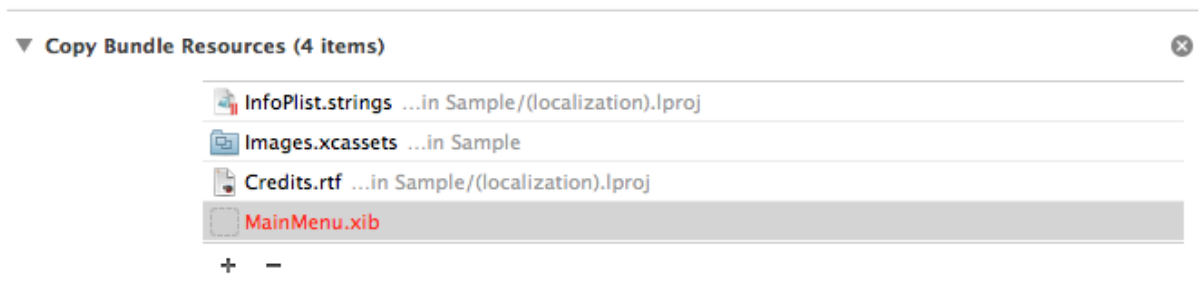
On the Choose Frameworks screen, click "Add Other" in the lower left

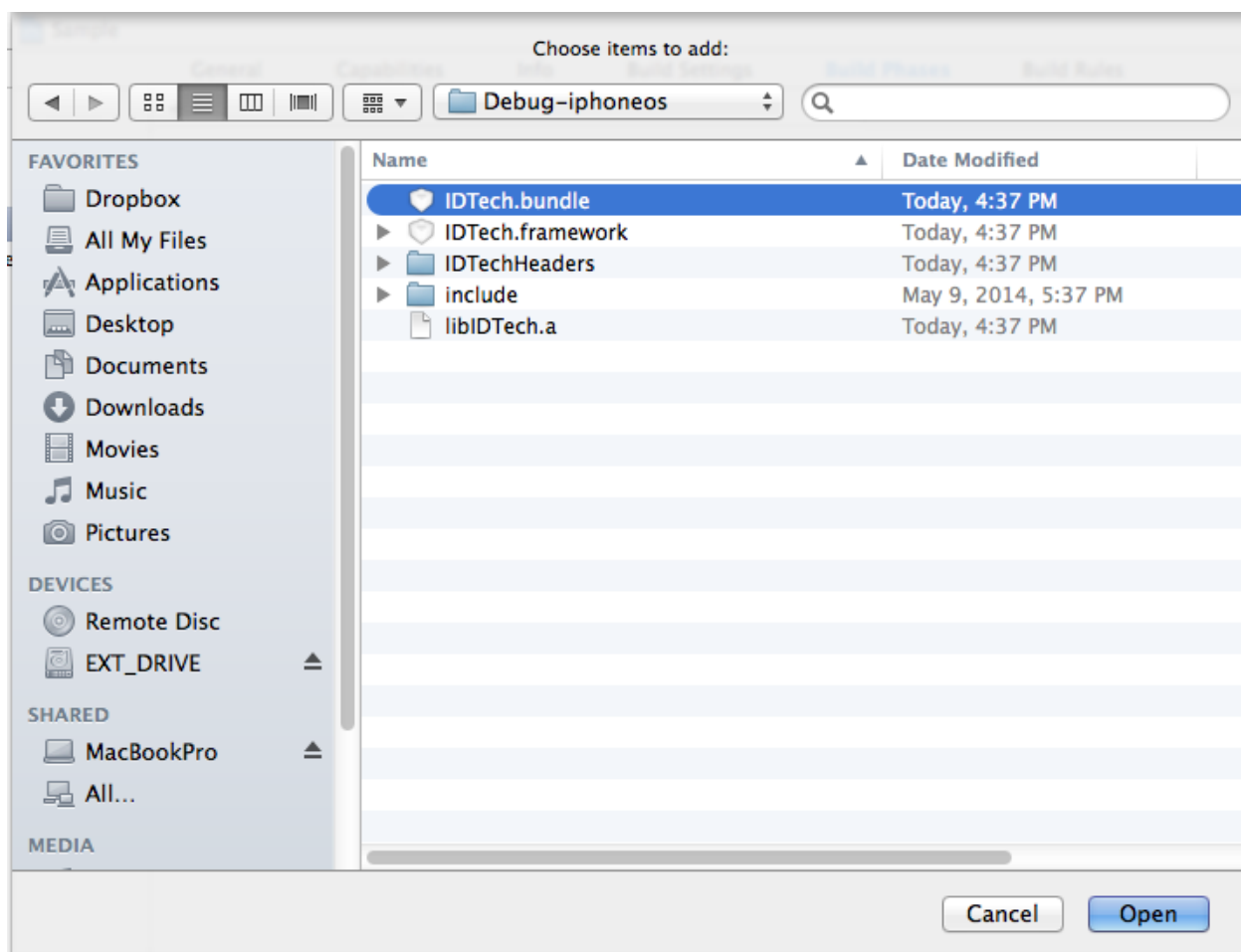
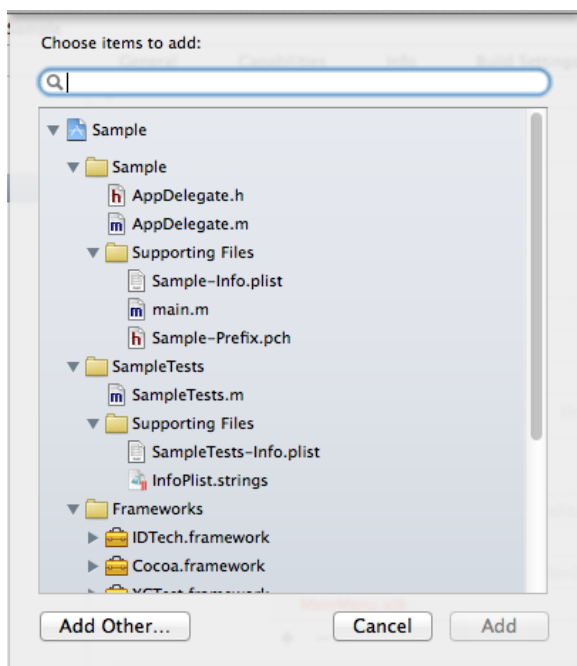


Navigate to the IDTech.framework folder (/Library/Frameworks/), and click "Open"

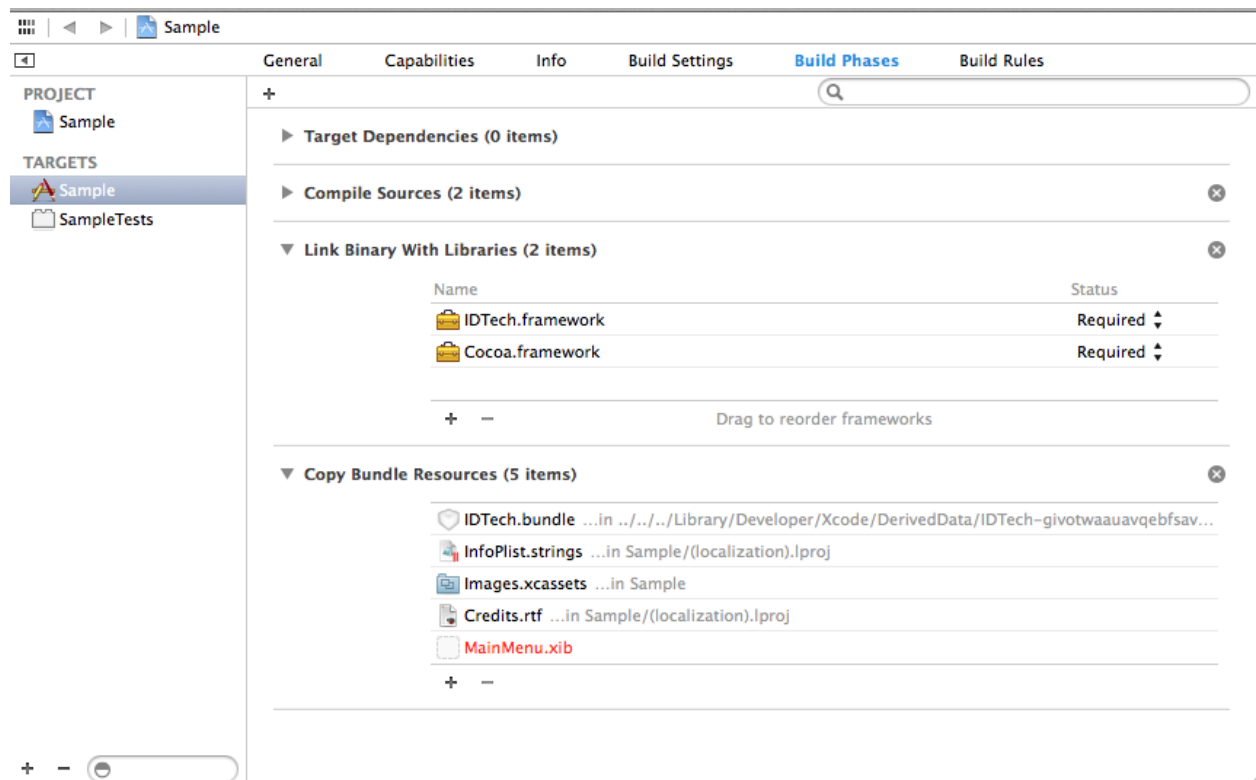


Link another library. Under Copy Bundle, click the Add (+) button, click "Add Other", navigate to and select the IDTech.bundle file and click "Open"





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



### 7.3 Add Import statements to utilize frameworks

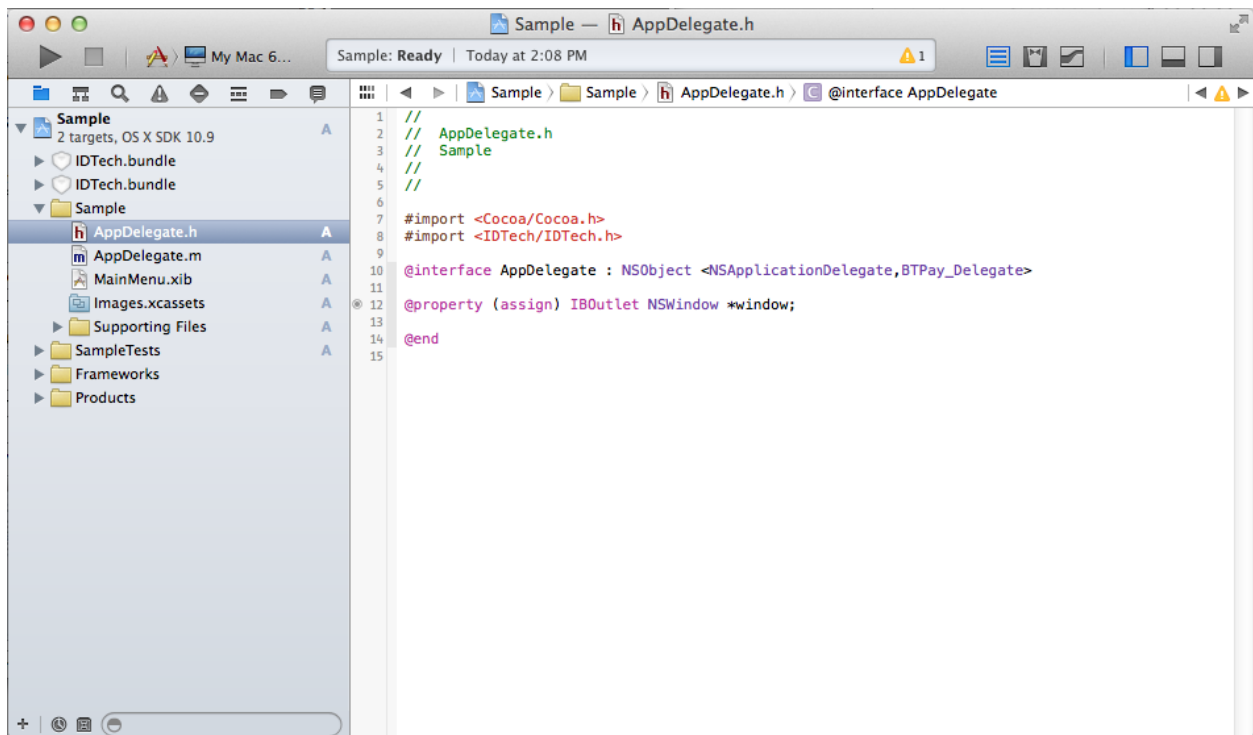
In the header files of the classes that will access the device, use import statement utilize the frameworks:

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

### 7.4 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 AppDelegate : NSObject <UIApplicationDelegate, IDT_BTPay_Delegate>
```



## 7.5 Implement any/all of the optional delegate protocols used to receive data from IDTech framework:

```

-(void) deviceConnected{
}

-(void) deviceDisconnected{
}

- (void) swipeMSRData:(IDTMSRData*) cardData{
}

- (void) pinpadData:(NSData*)value keySN:(NSData*)KSN event:(EVENT_PINPAD_Types)event{
}

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

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

```

## 7.6 Call the Singleton instance of the IDT\_BTPay framework object:

To use a device, a Singleton instance has been established in the [IDT\\_BTPay](#) class. To utilize the delegate protocols, best practices would be initialize the connection by setting the delegate with the singleton instance. Before calling the singleton instance for the first time, you must specify the device and connection method (USB/Bluetooth) with `IDT_BTPay::setDeviceType:()` or `IDT_UniPay::setDeviceType:()`.

```

typedef enum{
    IDT_DEVICE_BTPay_OSX_BT = 1,
    IDT_DEVICE_BTPay_OSX_USB
}IDT_Device_Types;

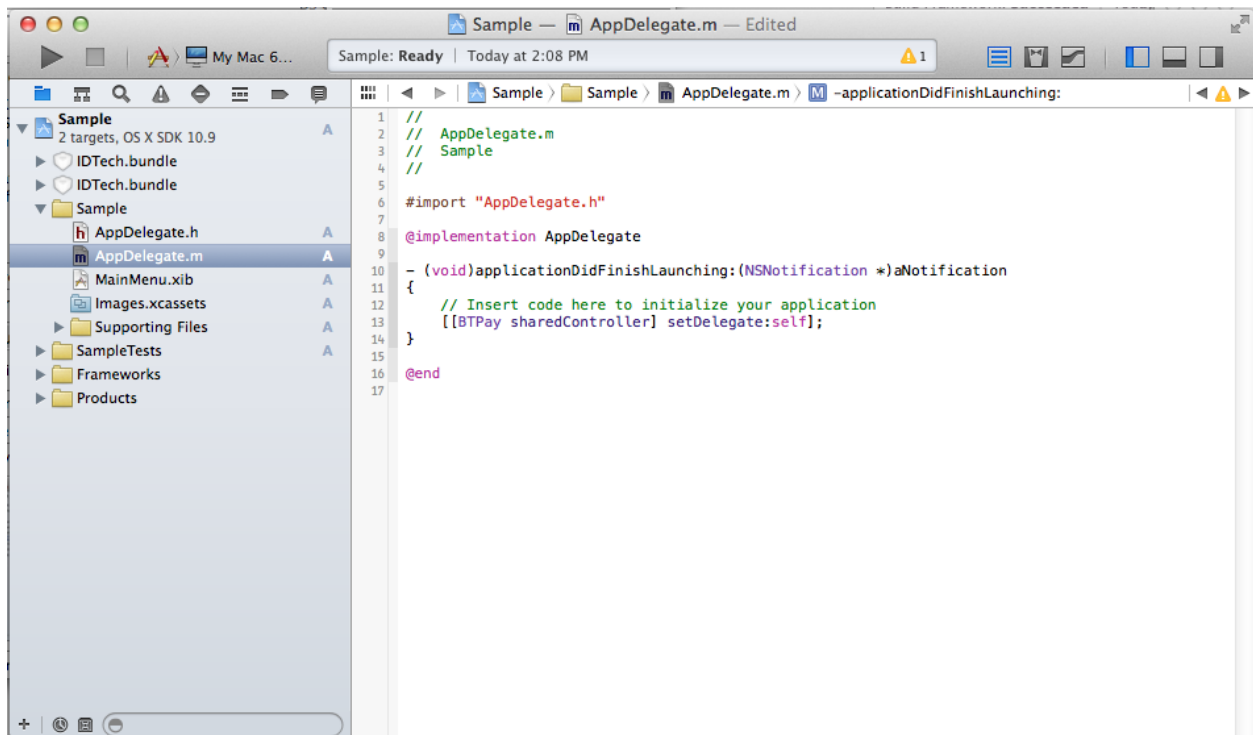
- (void)applicationDidFinishLaunching:(NSNotification *)aNotification
{
    // Insert code here to initialize your application
    [IDT_BTPay setDeviceType:IDT_DEVICE_BTPay_OSX_USB]; //We want to connect to BTPay 200 over
    USB-HID
}

```

```

    [[IDT_BTPay sharedController] setDelegate:self];
}

```



## 7.7 Sample Project Tutorial

Using Xcode 5.0+, we will create a sample project that will interface with the BTPay200 on OSX over USB or BT and will perform the following activities:

- Execute a beep
- Display unsecured message
- Request numeric input by passing secure message
- Turn on/off MSR reader and perform a card capture
- Connect/disconnect from device
- Request a PIN and capture KSN

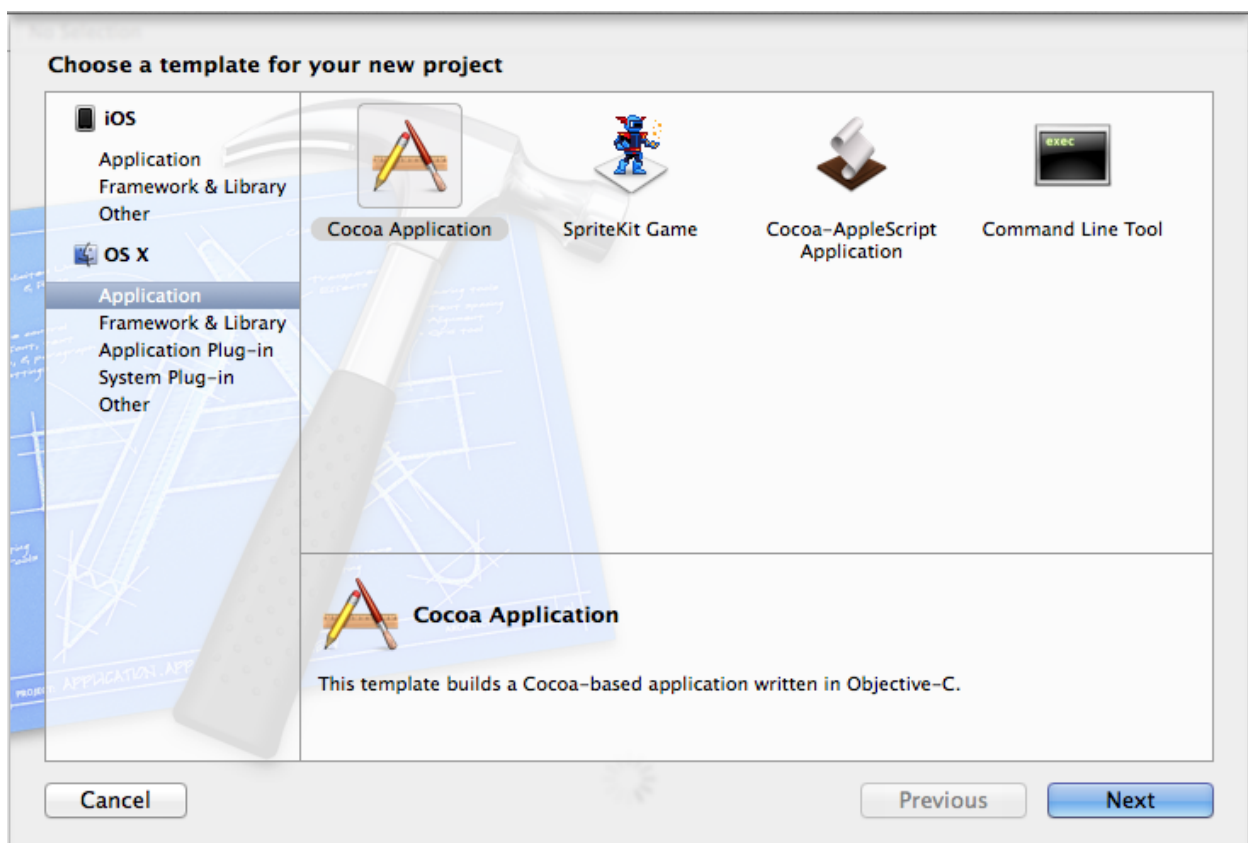
Protocol Delegates:

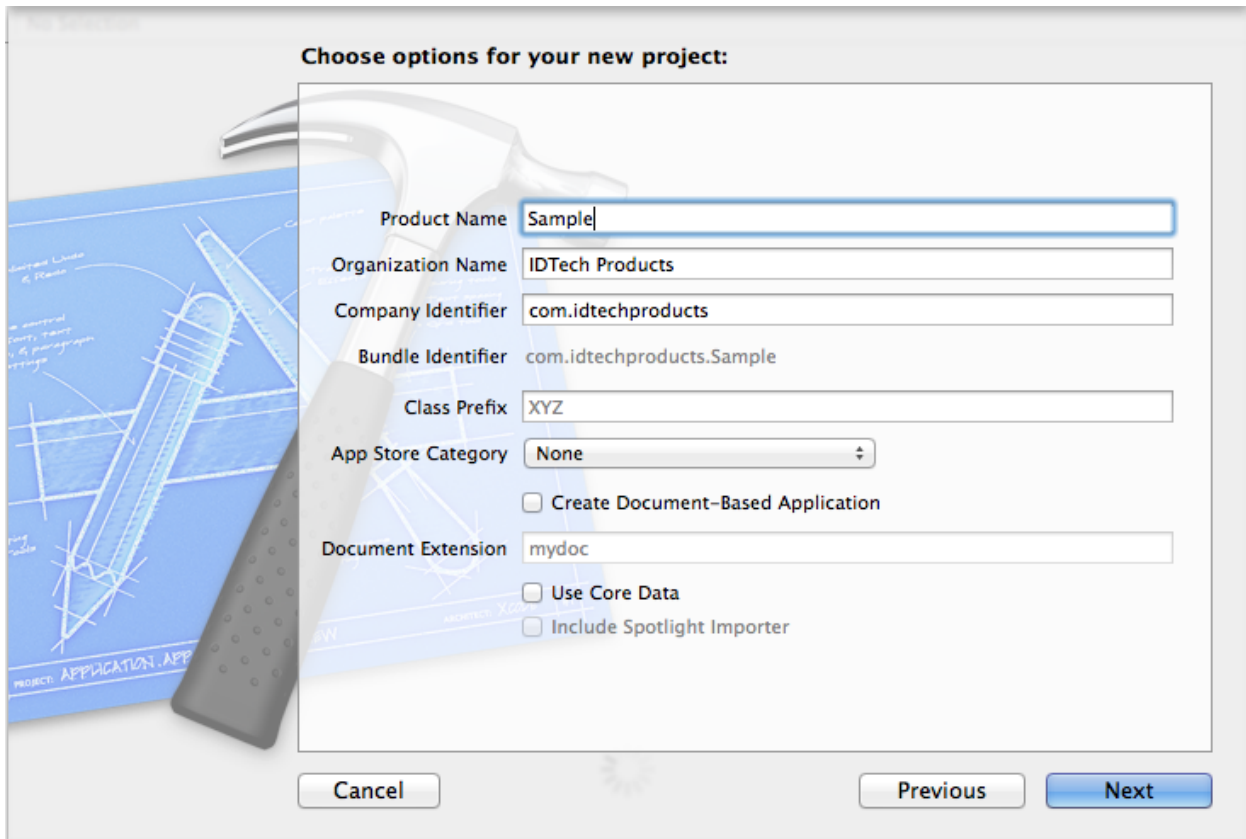
- Protocol to report unsolicited card swipes
- Protocol to report captured PIN
- Protocol to report device connected
- Protocol to report device disconnected
- Protocol to report all incoming and outgoing serial data



## 7.7.1 Step 1: Create New Project

Create a new Cocoa Application in Xcode





### 7.7.2 Step 2: Import Frameworks

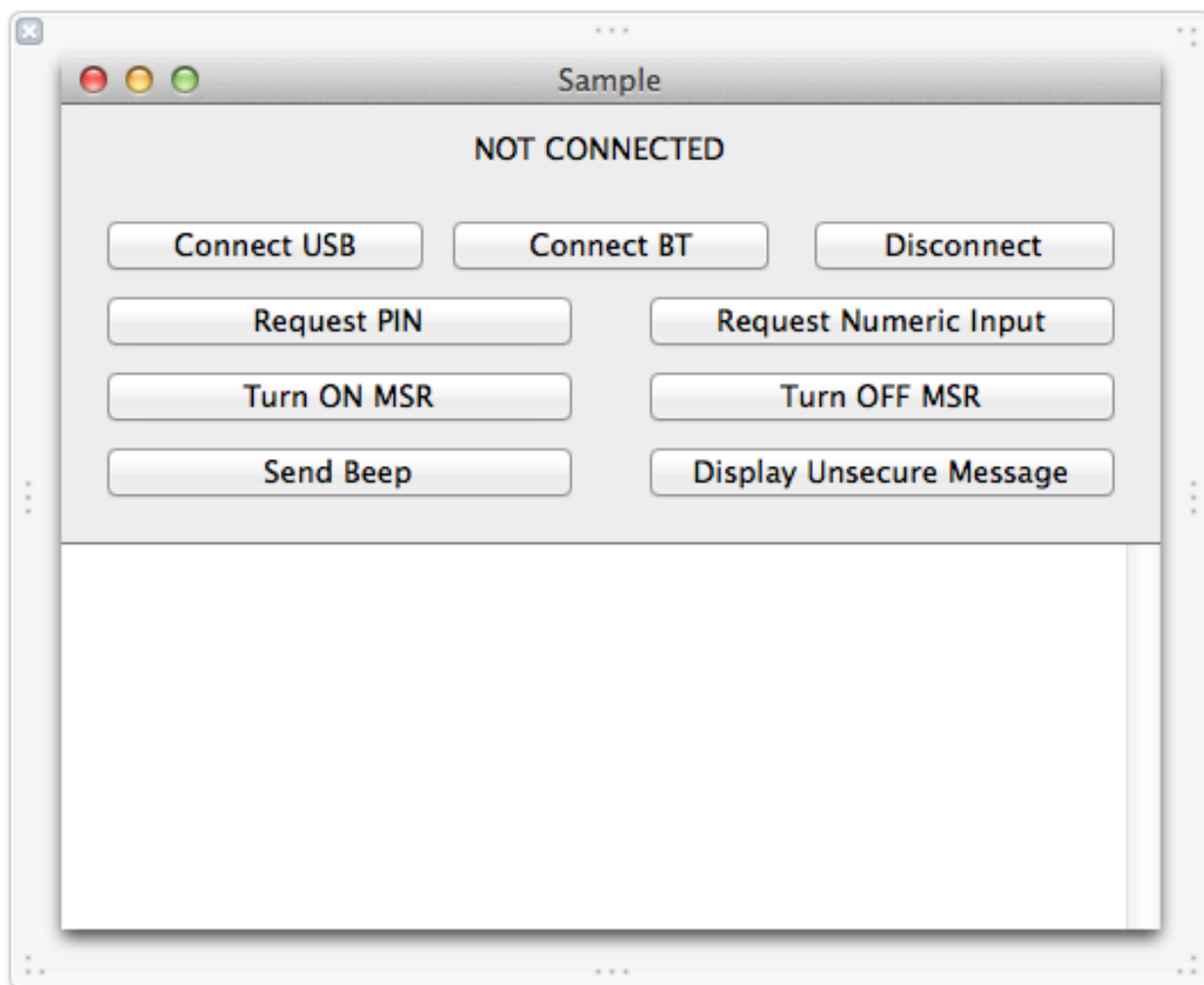
[Import the necessary framework/libraries](#)

### 7.7.3 Step 3: Design Interface

Design the User Interface by editing the OSX .xib file

Open your main .xib file, select the Window element and add items to so it contains the following buttons/fields:

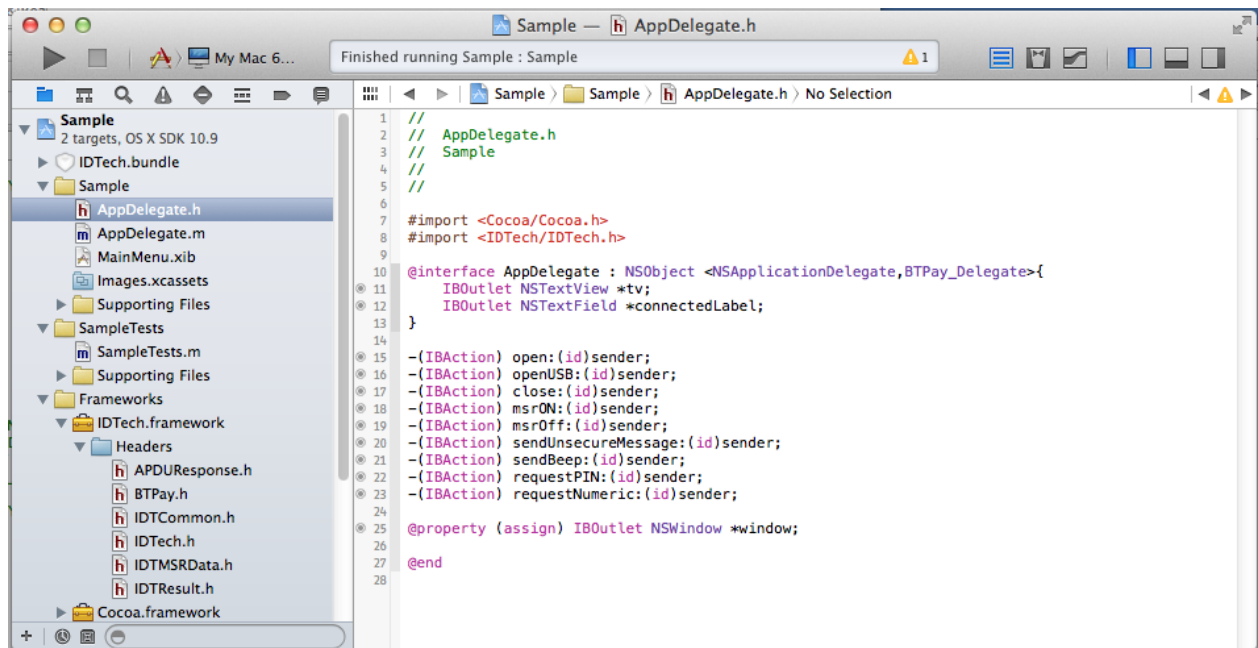
- Add a label to the top that will signify connection/disconnection status.
- Add a text view to communicate data from the BTPay200.
- Add buttons to execute the following functions:
  - Connect to device over USB-HID
  - Connect to device over BT
  - Disconnect from device
  - Request PIN
  - Request Numeric Input
  - Turn ON MSR
  - Turn OFF MSR
  - Send Unsecure Message
  - Send Beep



#### 7.7.4 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 text view and link it as a Referencing Outlet to the UITextView on the storyboard
- Create an IBOutlet for the label and link it as a Referencing Outlet to the NSTextField on the storyboard
- Create the 9 IBAction for the buttons, and link them to the selector on the xib buttons



```

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

@interface AppDelegate : NSObject <NSApplicationDelegate, IDT_BTPay_Delegate>{
    IBOutlet NSTextView *tv;
    IBOutlet NSTextField *connectedLabel;
}

-(IBAction) open:(id)sender;
-(IBAction) openUSB:(id)sender;
-(IBAction) close:(id)sender;
-(IBAction) msrON:(id)sender;
-(IBAction) msrOff:(id)sender;
-(IBAction) sendUnsecureMessage:(id)sender;
-(IBAction) device_sendBeep:(id)sender;
-(IBAction) requestPIN:(id)sender;
-(IBAction) requestNumeric:(id)sender;

@property (assign) IBOutlet NSWindow *window;

@end

```

## XIB Source Code

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.Cocoa.XIB" version="3.0" toolsVersion="5056" systemVersion="
13D65" targetRuntime="MacOSX.Cocoa" propertyAccessControl="none" useAutolayout="YES">
<dependencies>
<plugin identifier="com.apple.InterfaceBuilder.CocoaPlugin" version="5056"/>
</dependencies>
<objects>
<customObject id="-2" userLabel="File's Owner" customClass="NSApplication">
<connections>
<outlet property="delegate" destination="Voe-Tx-rLC" id="GzC-gU-4Uq"/>
</connections>
</customObject>
<customObject id="-1" userLabel="First Responder" customClass="FirstResponder"/>
<customObject id="-3" userLabel="Application"/>
<customObject id="Voe-Tx-rLC" customClass="AppDelegate">
<connections>
<outlet property="connectedLabel" destination="6lL-ua-Kka" id="Kyp-9t-eR6"/>
<outlet property="tv" destination="zQp-Eg-vR3" id="wgN-yQ-XhJ"/>
<outlet property="window" destination="QvC-M9-y7g" id="gIp-Ho-8D9"/>
</connections>
</customObject>
<customObject id="YLy-65-1bz" customClass="NSFontManager"/>
<menu title="Main Menu" systemMenu="main" id="AYu-sK-qS6">
<items>
<menuItem title="Sample" id="1Xt-HY-uBw">
<modifierMask key="keyEquivalentModifierMask"/>
<menu key="submenu" title="Sample" systemMenu="apple" id="uQy-DD-JDr">
<items>

```

```

        <menuItem title="About Sample" id="5kV-Vb-QxS">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="orderFrontStandardAboutPanel:" target="-1" id="
Exp-CZ-Vem"/>
            </connections>
        </menuItem>
        <menuItem isSeparatorItem="YES" id="VOq-y0-SEH"/>
        <menuItem title="Preferences..." keyEquivalent="," id="BOF-NM-1cW"/>
        <menuItem isSeparatorItem="YES" id="wFC-TO-SCJ"/>
        <menuItem title="Services" id="NMo-om-nkz">
            <modifierMask key="keyEquivalentModifierMask"/>
            <menu key="submenu" title="Services" systemMenu="services" id="hz9-B4-Xy5"/
>
        </menuItem>
        <menuItem isSeparatorItem="YES" id="4je-JR-u6R"/>
        <menuItem title="Hide Sample" keyEquivalent="h" id="Olw-nP-bQN">
            <connections>
                <action selector="hide:" target="-1" id="PnN-Uc-m68"/>
            </connections>
        </menuItem>
        <menuItem title="Hide Others" keyEquivalent="h" id="Vdr-fp-Xz0">
            <modifierMask key="keyEquivalentModifierMask" option="YES" command="YES"/>
            <connections>
                <action selector="hideOtherApplications:" target="-1" id="VT4-aY-XCT"/>
            </connections>
        </menuItem>
        <menuItem title="Show All" id="Kd2-mp-pUS">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="unhideAllApplications:" target="-1" id="Dhg-Le-xox"/>
            </connections>
        </menuItem>
        <menuItem isSeparatorItem="YES" id="kCx-OE-vgT"/>
        <menuItem title="Quit Sample" keyEquivalent="q" id="4sb-4s-VLi">
            <connections>
                <action selector="terminate:" target="-1" id="Te7-pn-YzF"/>
            </connections>
        </menuItem>
    </items>
</menu>
</menuItem>
<menuItem title="File" id="dMs-cI-mzQ">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="File" id="bib-Uj-vzu">
        <items>
            <menuItem title="New" keyEquivalent="n" id="Was-JA-tGl">
                <connections>
                    <action selector="newDocument:" target="-1" id="4Si-XN-c54"/>
                </connections>
            </menuItem>
            <menuItem title="Open..." keyEquivalent="o" id="IAo-SY-fd9">
                <connections>
                    <action selector="openDocument:" target="-1" id="bVn-NM-KNZ"/>
                </connections>
            </menuItem>
            <menuItem title="Open Recent" id="tXI-mr-wws">
                <modifierMask key="keyEquivalentModifierMask"/>
                <menu key="submenu" title="Open Recent" systemMenu="recentDocuments" id="
oas-Oc-fiZ">
                    <items>
                        <menuItem title="Clear Menu" id="vNY-rz-j42">
                            <modifierMask key="keyEquivalentModifierMask"/>
                            <connections>
                                <action selector="clearRecentDocuments:" target="-1" id="
Daa-9d-B3U"/>
                            </connections>
                        </menuItem>
                    </items>
                </menu>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="m54-Is-iLE"/>
            <menuItem title="Close" keyEquivalent="w" id="DVo-aG-piG">
                <connections>
                    <action selector="performClose:" target="-1" id="HmO-Is-i7Q"/>
                </connections>
            </menuItem>
            <menuItem title="Save..." keyEquivalent="s" id="pxx-59-PXV">
                <connections>
                    <action selector="saveDocument:" target="-1" id="teZ-XB-qJY"/>
                </connections>
            </menuItem>
            <menuItem title="Save As..." keyEquivalent="S" id="Bw7-FT-i3A">
                <connections>
                    <action selector="saveDocumentAs:" target="-1" id="mDf-zr-I0C"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>

```

```

        <menuItem title="Revert to Saved" id="KaW-ft-85H">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="revertDocumentToSaved:" target="-1" id="iJ3-Pv-kwq"/>
            </connections>
        </menuItem>
        <menuItem isSeparatorItem="YES" id="aJh-i4-bef"/>
        <menuItem title="Page Setup..." keyEquivalent="P" id="qIS-W8-SiK">
            <modifierMask key="keyEquivalentModifierMask" shift="YES" command="YES"/>
            <connections>
                <action selector="runPageLayout:" target="-1" id="Din-rz-gC5"/>
            </connections>
        </menuItem>
        <menuItem title="Print..." keyEquivalent="p" id="aTl-lu-JFS">
            <connections>
                <action selector="print:" target="-1" id="qaZ-4w-aoO"/>
            </connections>
        </menuItem>
    </items>
</menu>
</menuItem>
<menuItem title="Edit" id="5QF-Oa-p0T">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Edit" id="W48-6f-4D1">
        <items>
            <menuItem title="Undo" keyEquivalent="z" id="dRJ-4n-Yzg">
                <connections>
                    <action selector="undo:" target="-1" id="M6e-cu-g7V"/>
                </connections>
            </menuItem>
            <menuItem title="Redo" keyEquivalent="Z" id="6dh-zS-Vam">
                <connections>
                    <action selector="redo:" target="-1" id="oIA-Rs-6OD"/>
                </connections>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="WRV-NI-Exz"/>
            <menuItem title="Cut" keyEquivalent="x" id="uRl-iY-unG">
                <connections>
                    <action selector="cut:" target="-1" id="YJe-68-I9s"/>
                </connections>
            </menuItem>
            <menuItem title="Copy" keyEquivalent="c" id="x3v-GG-iWU">
                <connections>
                    <action selector="copy:" target="-1" id="GlF-GL-Joy"/>
                </connections>
            </menuItem>
            <menuItem title="Paste" keyEquivalent="v" id="gVA-U4-sdL">
                <connections>
                    <action selector="paste:" target="-1" id="UvS-8e-Qdg"/>
                </connections>
            </menuItem>
            <menuItem title="Paste and Match Style" keyEquivalent="V" id="WeT-3V-zwk">
                <modifierMask key="keyEquivalentModifierMask" option="YES" command="YES"/>
                <connections>
                    <action selector="pasteAsPlainText:" target="-1" id="cEh-KX-wJQ"/>
                </connections>
            </menuItem>
            <menuItem title="Delete" id="pa3-QI-u2k">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="delete:" target="-1" id="OMk-Ml-PaM"/>
                </connections>
            </menuItem>
            <menuItem title="Select All" keyEquivalent="a" id="RuW-6m-B2m">
                <connections>
                    <action selector="selectAll:" target="-1" id="VNm-Mi-diN"/>
                </connections>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="uyl-h8-X02"/>
            <menuItem title="Find" id="4EN-yA-p0u">
                <modifierMask key="keyEquivalentModifierMask"/>
                <menu key="submenu" title="Find" id="lb7-l0-nxx">
                    <items>
                        <menuItem title="Find..." tag="1" keyEquivalent="f" id="Xz5-n4-00W">
                            <connections>
                                <action selector="performFindPanelAction:" target="-1" id="
cD7-Qs-BN4"/>
                            </connections>
                        </menuItem>
                        <menuItem title="Find and Replace..." tag="12" keyEquivalent="f" id="
Yey-JH-Tfz">
                            <modifierMask key="keyEquivalentModifierMask" option="YES"
command="YES"/>
                            <connections>
                                <action selector="performFindPanelAction:" target="-1" id="
WD3-Gg-5AJ"/>
                            </connections>
                    </items>
                </menu>
            </menuItem>
        </items>
    </menu>
</menuItem>

```

```

        </menuItem>
        <menuItem title="Find Next" tag="2" keyEquivalent="g" id="
q09-ft-Sye">
            <connections>
                <action selector="performFindPanelAction:" target="-1" id="
NDo-RZ-v9R"/>
            </connections>
        </menuItem>
        <menuItem title="Find Previous" tag="3" keyEquivalent="G" id="
OwM-mh-QMV">
            <connections>
                <action selector="performFindPanelAction:" target="-1" id="
HOH-sY-3ay"/>
            </connections>
        </menuItem>
        <menuItem title="Use Selection for Find" tag="7" keyEquivalent="e"
id="buJ-ug-pKt">
            <connections>
                <action selector="performFindPanelAction:" target="-1" id="
U76-nv-p5D"/>
            </connections>
        </menuItem>
        <menuItem title="Jump to Selection" keyEquivalent="j" id="
S0p-oC-mLd">
            <connections>
                <action selector="centerSelectionInVisibleArea:" target="-1
" id="IOG-6D-g5B"/>
            </connections>
        </menuItem>
    </items>
</menu>
</menuItem>
<menuItem title="Spelling and Grammar" id="Dv1-io-Yv7">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Spelling" id="3IN-sU-3Bg">
        <items>
            <menuItem title="Show Spelling and Grammar" keyEquivalent=":" id="
HFo-cy-zxI">
                <connections>
                    <action selector="showGuessPanel:" target="-1" id="
vFj-Ks-hy3"/>
                </connections>
            </menuItem>
            <menuItem title="Check Document Now" keyEquivalent=";" id="
hz2-CU-CR7">
                <connections>
                    <action selector="checkSpelling:" target="-1" id="
fz7-VC-reM"/>
                </connections>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="bNw-od-mp5"/>
            <menuItem title="Check Spelling While Typing" id="rbD-Rh-wIN">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="toggleContinuousSpellChecking:" target="
-1" id="7w6-Qz-0kB"/>
                </connections>
            </menuItem>
            <menuItem title="Check Grammar With Spelling" id="mK6-2p-4JG">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="toggleGrammarChecking:" target="-1" id="
muD-Qn-j4w"/>
                </connections>
            </menuItem>
            <menuItem title="Correct Spelling Automatically" id="78Y-hA-62v">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="toggleAutomaticSpellingCorrection:"
target="-1" id="2lM-Qi-WAP"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>
<menuItem title="Substitutions" id="9ic-FL-obx">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Substitutions" id="FeM-D8-WVr">
        <items>
            <menuItem title="Show Substitutions" id="z6F-FW-3nz">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="orderFrontSubstitutionsPanel:" target="-1
" id="oku-mr-iSq"/>
                </connections>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="gPx-C9-uU0"/>
        </items>
    </menu>
</menuItem>

```

```

        <menuItem title="Smart Copy/Paste" id="9yt-4B-nSM">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="toggleSmartInsertDelete:" target="-1" id=
"3IJ-Se-DZD"/>
            </connections>
        </menuItem>
        <menuItem title="Smart Quotes" id="hQb-2v-fYv">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="toggleAutomaticQuoteSubstitution:" target
="-1" id="ptq-xd-QOA"/>
            </connections>
        </menuItem>
        <menuItem title="Smart Dashes" id="rgM-f4-ycn">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="toggleAutomaticDashSubstitution:" target=
"-1" id="oCt-pO-9gS"/>
            </connections>
        </menuItem>
        <menuItem title="Smart Links" id="cwL-P1-jid">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="toggleAutomaticLinkDetection:" target="-1
" id="Gip-E3-Fov"/>
            </connections>
        </menuItem>
        <menuItem title="Data Detectors" id="tRr-pd-lPS">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="toggleAutomaticDataDetection:" target="-1
" id="RlI-Nq-Kbl"/>
            </connections>
        </menuItem>
        <menuItem title="Text Replacement" id="HFQ-gK-NFA">
            <modifierMask key="keyEquivalentModifierMask"/>
            <connections>
                <action selector="toggleAutomaticTextReplacement:" target="
-1" id="DvP-Fe-Py6"/>
            </connections>
        </menuItem>
    </items>
</menu>
</menuItem>
<menuItem title="Transformations" id="2oI-Rn-ZJC">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Transformations" id="c8a-y6-VQd">
        <items>
            <menuItem title="Make Upper Case" id="vmV-6d-7jI">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="uppercaseWord:" target="-1" id="
sPh-Tk-edu"/>
                </connections>
            </menuItem>
            <menuItem title="Make Lower Case" id="d9M-CD-aMd">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="lowercaseWord:" target="-1" id="
iUZ-b5-hil"/>
                </connections>
            </menuItem>
            <menuItem title="Capitalize" id="UEZ-Bs-lqG">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="capitalizeWord:" target="-1" id="
26H-TL-nsh"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>
<menuItem title="Speech" id="xrE-MZ-jX0">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Speech" id="3rS-ZA-NoH">
        <items>
            <menuItem title="Start Speaking" id="Ynk-f8-clZ">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="startSpeaking:" target="-1" id="
654-Ng-kyl"/>
                </connections>
            </menuItem>
            <menuItem title="Stop Speaking" id="Oyz-dy-DGm">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>

```



```

        <action selector="stopSpeaking:" target="-1" id="dX8-6p-jy9
"/>
        </connections>
    </menuItem>
</items>
</menu>
</menuItem>
</items>
</menu>
</menuItem>
<menuItem title="Format" id="jxT-CU-nIS">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Format" id="GEO-Iw-cKr">
        <items>
            <menuItem title="Font" id="Gi5-1S-RQB">
                <modifierMask key="keyEquivalentModifierMask"/>
                <menu key="submenu" title="Font" systemMenu="font" id="aXa-aM-Jaq">
                    <items>
                        <menuItem title="Show Fonts" keyEquivalent="t" id="Q5e-8K-NDq">
                            <connections>
                                <action selector="orderFrontFontPanel:" target="YLy-65-1bz"
id="WHr-nq-2xA"/>
                            </connections>
                        </menuItem>
                        <menuItem title="Bold" tag="2" keyEquivalent="b" id="GB9-OM-e27">
                            <connections>
                                <action selector="addFontTrait:" target="YLy-65-1bz" id="
hqk-hr-sYV"/>
                            </connections>
                        </menuItem>
                        <menuItem title="Italic" tag="1" keyEquivalent="i" id="Vjx-xi-njq">
                            <connections>
                                <action selector="addFontTrait:" target="YLy-65-1bz" id="
IHV-OB-c03"/>
                            </connections>
                        </menuItem>
                        <menuItem title="Underline" keyEquivalent="u" id="WRG-CD-K1S">
                            <connections>
                                <action selector="underline:" target="-1" id="FYS-2b-JAY"/>
                            </connections>
                        </menuItem>
                        <menuItem isSeparatorItem="YES" id="5gT-KC-WSO"/>
                        <menuItem title="Bigger" tag="3" keyEquivalent="+" id="Ptp-SP-VEL">
                            <connections>
                                <action selector="modifyFont:" target="YLy-65-1bz" id="
Uc7-di-UnL"/>
                            </connections>
                        </menuItem>
                        <menuItem title="Smaller" tag="4" keyEquivalent="-" id="ild-Er-qSt"
>
                            <connections>
                                <action selector="modifyFont:" target="YLy-65-1bz" id="
HcX-Lf-eNd"/>
                            </connections>
                        </menuItem>
                        <menuItem isSeparatorItem="YES" id="kx3-Dk-x3B"/>
                        <menuItem title="Kern" id="jBQ-r6-VK2">
                            <modifierMask key="keyEquivalentModifierMask"/>
                            <menu key="submenu" title="Kern" id="tLD-Oa-oAM">
                                <items>
                                    <menuItem title="Use Default" id="GUa-eO-cwY">
                                        <modifierMask key="keyEquivalentModifierMask"/>
                                        <connections>
                                            <action selector="useStandardKerning:" target="
-1" id="6dk-9l-Ckg"/>
                                        </connections>
                                    </menuItem>
                                    <menuItem title="Use None" id="cDB-IK-hbR">
                                        <modifierMask key="keyEquivalentModifierMask"/>
                                        <connections>
                                            <action selector="turnOffKerning:" target="-1"
id="U8a-gz-Maa"/>
                                        </connections>
                                    </menuItem>
                                    <menuItem title="Tighten" id="46P-cB-AYj">
                                        <modifierMask key="keyEquivalentModifierMask"/>
                                        <connections>
                                            <action selector="tightenKerning:" target="-1"
id="hr7-Nz-8ro"/>
                                        </connections>
                                    </menuItem>
                                    <menuItem title="Loosen" id="ogc-rX-tC1">
                                        <modifierMask key="keyEquivalentModifierMask"/>
                                        <connections>
                                            <action selector="loosenKerning:" target="-1"
id="8i4-f9-FKE"/>
                                        </connections>
                                </items>
                            </menu>
                        </menuItem>
                    </items>
                </menu>
            </menuItem>
        </items>
    </menu>
</menuItem>

```

```

        </menuItem>
    </items>
</menu>
</menuItem>
<menuItem title="Ligatures" id="o6e-r0-MWg">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Ligatures" id="w0m-vy-SC9">
        <items>
            <menuItem title="Use Default" id="agt-UL-0e3">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="useStandardLigatures:" target=
=" -1" id="7uR-wd-Dx6"/>
                </connections>
            </menuItem>
            <menuItem title="Use None" id="J7y-lM-qPV">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="turnOffLigatures:" target="-1
" id="iX2-gA-Ilz"/>
                </connections>
            </menuItem>
            <menuItem title="Use All" id="xQD-lf-W4t">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="useAllLigatures:" target="-1"
id="KcB-kA-TuK"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>
<menuItem title="Baseline" id="OaQ-X3-Vso">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Baseline" id="ijk-EB-dga">
        <items>
            <menuItem title="Use Default" id="3Om-Ey-2VK">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="unscript:" target="-1" id="
0vZ-95-Ywn"/>
                </connections>
            </menuItem>
            <menuItem title="Superscript" id="Rqc-34-cIF">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="superscript:" target="-1" id=
"3qV-fo-wpU"/>
                </connections>
            </menuItem>
            <menuItem title="Subscript" id="IOS-gh-461">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="subscript:" target="-1" id="
Q6W-4W-IGz"/>
                </connections>
            </menuItem>
            <menuItem title="Raise" id="2h7-ER-AoG">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="raiseBaseline:" target="-1"
id="4sk-31-7Q9"/>
                </connections>
            </menuItem>
            <menuItem title="Lower" id="ltx-W0-xDw">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="lowerBaseline:" target="-1"
id="OF1-bc-KW4"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>
<menuItem isSeparatorItem="YES" id="Ndw-q3-faq"/>
<menuItem title="Show Colors" keyEquivalent="C" id="bgn-CT-cEk">
    <connections>
        <action selector="orderFrontColorPanel:" target="-1" id="
mSX-Xz-DV3"/>
    </connections>
</menuItem>
<menuItem isSeparatorItem="YES" id="iMs-zA-UFJ"/>
<menuItem title="Copy Style" keyEquivalent="c" id="5Vv-lz-BsD">
    <modifierMask key="keyEquivalentModifierMask" option="YES"
command="YES"/>
    <connections>
        <action selector="copyFont:" target="-1" id="GJO-xA-L4q"/>
    </connections>

```

```

        </connections>
    </menuItem>
    <menuItem title="Paste Style" keyEquivalent="v" id="vKC-jM-MkH">
        <modifierMask key="keyEquivalentModifierMask" option="YES"

command="YES"/>

        <connections>
            <action selector="pasteFont:" target="-1" id="JfD-CL-le0"/>
        </connections>
    </menuItem>
</items>
</menu>
</menuItem>
<menuItem title="Text" id="Fal-I4-PZk">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Text" id="d9c-me-L2H">
        <items>
            <menuItem title="Align Left" keyEquivalent="{ " id="ZM1-6Q-yy1">
                <connections>
                    <action selector="alignLeft:" target="-1" id="zUv-R1-uAa"/>
                </connections>
            </menuItem>
            <menuItem title="Center" keyEquivalent="|" id="VIY-Ag-zcb">
                <connections>
                    <action selector="alignCenter:" target="-1" id="spX-mk-kcS"

/>

                </connections>
            </menuItem>
            <menuItem title="Justify" id="J5U-5w-g23">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="alignJustified:" target="-1" id="

1jL-7U-jND"/>

                </connections>
            </menuItem>
            <menuItem title="Align Right" keyEquivalent="}" id="wb2-vD-lq4">
                <connections>
                    <action selector="alignRight:" target="-1" id="r48-bG-YeY"/

>

                </connections>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="4s2-GY-VfK"/>
            <menuItem title="Writing Direction" id="H1b-Si-o9J">
                <modifierMask key="keyEquivalentModifierMask"/>
                <menu key="submenu" title="Writing Direction" id="8mr-sm-Yjd">
                    <items>
                        <menuItem title="Paragraph" enabled="NO" id="Zv0-Gk-QUH

">

                            <modifierMask key="keyEquivalentModifierMask"/>
                        </menuItem>
                        <menuItem id="YGs-j5-SAR">
                            <string key="title"> Default</string>
                            <modifierMask key="keyEquivalentModifierMask"/>
                            <connections>
                                <action selector="

makeBaseWritingDirectionNatural:" target="-1" id="qtV-5e-UBP"/>
                            </connections>
                        </menuItem>
                        <menuItem id="Lbh-J2-qVU">
                            <string key="title"> Left to Right</string>
                            <modifierMask key="keyEquivalentModifierMask"/>
                            <connections>
                                <action selector="

makeBaseWritingDirectionLeftToRight:" target="-1" id="SOX-9S-QSf"/>
                            </connections>
                        </menuItem>
                        <menuItem id="jFq-tB-4Kx">
                            <string key="title"> Right to Left</string>
                            <modifierMask key="keyEquivalentModifierMask"/>
                            <connections>
                                <action selector="

makeBaseWritingDirectionRightToLeft:" target="-1" id="5fk-qB-AqJ"/>
                            </connections>
                        </menuItem>
                        <menuItem isSeparatorItem="YES" id="swp-gr-a21"/>
                        <menuItem title="Selection" enabled="NO" id="cqv-fj-IhA

">

                            <modifierMask key="keyEquivalentModifierMask"/>
                        </menuItem>
                        <menuItem id="Nop-cj-93Q">
                            <string key="title"> Default</string>
                            <modifierMask key="keyEquivalentModifierMask"/>
                            <connections>
                                <action selector="

makeTextWritingDirectionNatural:" target="-1" id="lPI-Se-ZHp"/>
                            </connections>
                        </menuItem>
                        <menuItem id="BgM-ve-c93">

```

```

        <string key="title">    Left to Right</string>
        <modifierMask key="keyEquivalentModifierMask"/>
        <connections>
            <action selector="
makeTextWritingDirectionLeftToRight:" target="-1" id="caW-Bv-w94"/>
        </connections>
    </menuItem>
    <menuItem id="RB4-Sm-HuC">
        <string key="title">    Right to Left</string>
        <modifierMask key="keyEquivalentModifierMask"/>
        <connections>
            <action selector="
makeTextWritingDirectionRightToLeft:" target="-1" id="EXD-6r-ZUu"/>
        </connections>
    </menuItem>
</items>
</menu>
</menuItem>
<menuItem isSeparatorItem="YES" id="fKy-g9-lgm"/>
<menuItem title="Show Ruler" id="vLm-3I-IUL">
    <modifierMask key="keyEquivalentModifierMask"/>
    <connections>
        <action selector="toggleRuler:" target="-1" id="FOx-HJ-KwY"
/>
    </connections>
</menuItem>
<menuItem title="Copy Ruler" keyEquivalent="c" id="MkV-Pr-PK5">
    <modifierMask key="keyEquivalentModifierMask" control="YES"
command="YES"/>
    <connections>
        <action selector="copyRuler:" target="-1" id="7li-fW-3W2"/>
    </connections>
</menuItem>
<menuItem title="Paste Ruler" keyEquivalent="v" id="LVM-kO-fVI">
    <modifierMask key="keyEquivalentModifierMask" control="YES"
command="YES"/>
    <connections>
        <action selector="pasteRuler:" target="-1" id="cSh-wd-qM2"/>
    </connections>
</menuItem>
</items>
</menu>
</menuItem>
</items>
</menu>
</menuItem>
<menuItem title="View" id="H8h-7b-M4v">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="View" id="HyV-fh-RgO">
        <items>
            <menuItem title="Show Toolbar" keyEquivalent="t" id="snW-S8-Cw5">
                <modifierMask key="keyEquivalentModifierMask" option="YES" command="YES"/>
                <connections>
                    <action selector="toggleToolbarShown:" target="-1" id="BXY-wc-z0C"/>
                </connections>
            </menuItem>
            <menuItem title="Customize Toolbar..." id="lUK-8n-QPP">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="runToolbarCustomizationPalette:" target="-1" id="
pQI-g3-MTW"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>
<menuItem title="Window" id="aUF-d1-5bR">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Window" systemMenu="window" id="Td7-aD-5lo">
        <items>
            <menuItem title="Minimize" keyEquivalent="m" id="OY7-WF-poV">
                <connections>
                    <action selector="performMiniaturize:" target="-1" id="VwT-WD-YPe"/>
                </connections>
            </menuItem>
            <menuItem title="Zoom" id="R4o-n2-Eq4">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="performZoom:" target="-1" id="D1l-cC-cCs"/>
                </connections>
            </menuItem>
            <menuItem isSeparatorItem="YES" id="eu3-7i-yIM"/>
            <menuItem title="Bring All to Front" id="LE2-aR-0XJ">
                <modifierMask key="keyEquivalentModifierMask"/>
                <connections>
                    <action selector="arrangeInFront:" target="-1" id="DRN-fu-gQh"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>

```

```

        </connections>
    </menuItem>
</items>
</menu>
</menuItem>
<menuItem title="Help" id="wpr-3q-Mcd">
    <modifierMask key="keyEquivalentModifierMask"/>
    <menu key="submenu" title="Help" systemMenu="help" id="F2S-fz-NVQ">
        <items>
            <menuItem title="Sample Help" keyEquivalent="?" id="FKE-Sm-Kum">
                <connections>
                    <action selector="showHelp:" target="-1" id="y7X-2Q-9no"/>
                </connections>
            </menuItem>
        </items>
    </menu>
</menuItem>
</items>
</menu>
<window title="Sample" allowsToolTipsWhenApplicationIsInactive="NO" autorecalculatesKeyViewLoop="NO"
    releasedWhenClosed="NO" animationBehavior="default" id="QvC-M9-y7g">
    <windowStyleMask key="styleMask" titled="YES" closable="YES" miniaturizable="YES" resizable="YES"/>
    <windowPositionMask key="initialPositionMask" leftStrut="YES" rightStrut="YES" topStrut="YES"
    bottomStrut="YES"/>
    <rect key="contentRect" x="335" y="390" width="480" height="360"/>
    <rect key="screenRect" x="0.0" y="0.0" width="2560" height="1418"/>
    <view key="contentView" id="EiT-Mj-1SZ">
        <rect key="frame" x="0.0" y="0.0" width="480" height="360"/>
        <autoresizingMask key="autoresizingMask"/>
        <subviews>
            <button verticalHuggingPriority="750" fixedFrame="YES"
    translatesAutoresizingMaskIntoConstraints="NO" id="fP3-if-jat">
                <rect key="frame" x="14" y="248" width="215" height="32"/>
                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
                <buttonCell key="cell" type="push" title="Request PIN" bezelStyle="rounded"
    alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="jd9-kg-zCu">
                    <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
    />
                <font key="font" metaFont="system"/>
            </buttonCell>
            <connections>
                <action selector="requestPIN:" target="Voe-Tx-rLC" id="wOM-cD-8jq"/>
            </connections>
        </button>
        <button verticalHuggingPriority="750" fixedFrame="YES"
    translatesAutoresizingMaskIntoConstraints="NO" id="oCl-dv-MvY">
                <rect key="frame" x="251" y="248" width="215" height="32"/>
                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
                <buttonCell key="cell" type="push" title="Request Numeric Input" bezelStyle="rounded"
    alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="u8S-dA-oKX">
                    <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
    />
                <font key="font" metaFont="system"/>
            </buttonCell>
            <connections>
                <action selector="requestNumeric:" target="Voe-Tx-rLC" id="n3g-QJ-2mI"/>
            </connections>
        </button>
        <button verticalHuggingPriority="750" fixedFrame="YES"
    translatesAutoresizingMaskIntoConstraints="NO" id="YVv-JZ-kDW">
                <rect key="frame" x="14" y="182" width="215" height="32"/>
                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
                <buttonCell key="cell" type="push" title="Send Beep" bezelStyle="rounded" alignment
    ="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="MoR-Fk-Ylk">
                    <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
    />
                <font key="font" metaFont="system"/>
            </buttonCell>
            <connections>
                <action selector="device_sendBeep:" target="Voe-Tx-rLC" id="pkf-bM-tiZ"/>
            </connections>
        </button>
        <button verticalHuggingPriority="750" fixedFrame="YES"
    translatesAutoresizingMaskIntoConstraints="NO" id="Ulm-zo-okz">
                <rect key="frame" x="14" y="281" width="150" height="32"/>
                <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
                <buttonCell key="cell" type="push" title="Connect USB" bezelStyle="rounded"
    alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="INq-yP-O0C">
                    <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
    />
                <font key="font" metaFont="system"/>
            </buttonCell>
            <connections>
                <action selector="openUSB:" target="Voe-Tx-rLC" id="Upp-Ez-gdW"/>
            </connections>
        </button>

```

```

        <button verticalHuggingPriority="750" fixedFrame="YES"
translatesAutoresizingMaskIntoConstraints="NO" id="uu8-8M-luP">
        <rect key="frame" x="165" y="281" width="150" height="32"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
        <buttonCell key="cell" type="push" title="Connect BT" bezelStyle="rounded"
alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="40r-qG-goS">
        <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
/>

        <font key="font" metaFont="system"/>
    </buttonCell>
    <connections>
        <action selector="open:" target="Voe-Tx-rLC" id="RxQ-Wz-zZa"/>
    </connections>
</button>
<button verticalHuggingPriority="750" fixedFrame="YES"
translatesAutoresizingMaskIntoConstraints="NO" id="fyX-Tt-Mgs">
    <rect key="frame" x="323" y="281" width="143" height="32"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
    <buttonCell key="cell" type="push" title="Disconnect" bezelStyle="rounded"
alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="L4o-7D-FdR">
        <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
/>

        <font key="font" metaFont="system"/>
    </buttonCell>
    <connections>
        <action selector="close:" target="Voe-Tx-rLC" id="G8d-Fp-AYc"/>
    </connections>
</button>
<button verticalHuggingPriority="750" fixedFrame="YES"
translatesAutoresizingMaskIntoConstraints="NO" id="e3I-N5-e52">
    <rect key="frame" x="251" y="182" width="215" height="32"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
    <buttonCell key="cell" type="push" title="Display Unsecure Message" bezelStyle="
rounded" alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="Z8B-nQ-H1W">
        <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
/>

        <font key="font" metaFont="system"/>
    </buttonCell>
    <connections>
        <action selector="sendUnsecureMessage:" target="Voe-Tx-rLC" id="IsV-Sr-pBQ"/>
    </connections>
</button>
<button verticalHuggingPriority="750" fixedFrame="YES"
translatesAutoresizingMaskIntoConstraints="NO" id="mFU-70-yVq">
    <rect key="frame" x="14" y="215" width="215" height="32"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
    <buttonCell key="cell" type="push" title="Turn ON MSR" bezelStyle="rounded"
alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="g6f-rC-Fuc">
        <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
/>

        <font key="font" metaFont="system"/>
    </buttonCell>
    <connections>
        <action selector="msrON:" target="Voe-Tx-rLC" id="6F5-yF-EMH"/>
    </connections>
</button>
<button verticalHuggingPriority="750" fixedFrame="YES"
translatesAutoresizingMaskIntoConstraints="NO" id="bNX-Zn-7iS">
    <rect key="frame" x="251" y="215" width="215" height="32"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
    <buttonCell key="cell" type="push" title="Turn OFF MSR" bezelStyle="rounded"
alignment="center" borderStyle="border" imageScaling="proportionallyDown" inset="2" id="fga-VK-2g0">
        <behavior key="behavior" pushIn="YES" lightByBackground="YES" lightByGray="YES"
/>

        <font key="font" metaFont="system"/>
    </buttonCell>
    <connections>
        <action selector="msrOff:" target="Voe-Tx-rLC" id="w4u-Wo-svZ"/>
    </connections>
</button>
<textField horizontalHuggingPriority="251" verticalHuggingPriority="750" fixedFrame="
YES" translatesAutoresizingMaskIntoConstraints="NO" id="6lL-ua-Kka">
    <rect key="frame" x="39" y="333" width="393" height="17"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMinY="YES"/>
    <textFieldCell key="cell" scrollable="YES" lineBreakMode="clipping"
sendsActionOnEndEditing="YES" alignment="center" title="NOT CONNECTED" id="BX8-17-3Gb">
        <font key="font" metaFont="system"/>
        <color key="textColor" name="controlDarkShadowColor" catalog="System"
colorSpace="catalog"/>
        <color key="backgroundColor" name="textBackgroundColor" catalog="System"
colorSpace="catalog"/>
    </textFieldCell>
</textField>
<scrollView fixedFrame="YES" horizontalLineScroll="10" horizontalPageScroll="10"
verticalLineScroll="10" verticalPageScroll="10" hasHorizontalScroller="NO" usesPredominantAxisScrolling="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="F9R-eB-k2V">
    <rect key="frame" x="-1" y="-1" width="482" height="170"/>

```

```

        <autoresizingMask key="autoresizingMask"/>
        <clipView key="contentView" ambiguous="YES" misplaced="YES" id="Iqr-ND-gE9">
            <rect key="frame" x="1" y="1" width="465" height="168"/>
            <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"
        />

        <subviews>
            <textView ambiguous="YES" importsGraphics="NO" findStyle="panel"
continuousSpellChecking="YES" allowsUndo="YES" usesRuler="YES" usesFontPanel="YES" verticallyResizable="YES"
allowsNonContiguousLayout="YES" quoteSubstitution="YES" dashSubstitution="YES" spellingCorrection="YES"
smartInsertDelete="YES" id="zQp-Eg-vR3">
                <rect key="frame" x="0.0" y="0.0" width="465" height="168"/>
                <autoresizingMask key="autoresizingMask" widthSizable="YES"
heightSizable="YES"/>
                <color key="backgroundColor" white="1" alpha="1" colorSpace="
calibratedWhite"/>
                <size key="minSize" width="465" height="168"/>
                <size key="maxSize" width="482" height="10000000"/>
                <color key="insertionPointColor" white="0.0" alpha="1" colorSpace="
calibratedWhite"/>
                <size key="minSize" width="465" height="168"/>
                <size key="maxSize" width="482" height="10000000"/>
            </textView>
        </subviews>
        <color key="backgroundColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
    </clipView>
    <scroller key="horizontalScroller" hidden="YES" verticalHuggingPriority="750"
doubleValue="1" horizontal="YES" id="zbw-4a-ltJ">
        <rect key="frame" x="-100" y="-100" width="87" height="18"/>
        <autoresizingMask key="autoresizingMask"/>
    </scroller>
    <scroller key="verticalScroller" verticalHuggingPriority="750" doubleValue="1"
horizontal="NO" id="qkV-Lf-zbW">
        <rect key="frame" x="466" y="1" width="15" height="168"/>
        <autoresizingMask key="autoresizingMask"/>
    </scroller>
</scrollView>
</subviews>
</view>
</window>
</objects>
</document>

```

- Implement the button press methods and the delegates

```

//
// AppDelegate.m
// Sample
//

#import "AppDelegate.h"

@implementation AppDelegate

- (void)applicationDidFinishLaunching:(NSNotification *)aNotification
{
    // Insert code here to initialize your application
}

- (IBAction)open:(id)sender{
    [IDT_BTPay setDeviceType:IDT_DEVICE_BT_PAY_OSX_BT];
    [[IDT_BTPay sharedController] setDelegate:self];
}

- (IBAction)openUSB:(id)sender{
    [IDT_BTPay setDeviceType:IDT_DEVICE_BT_PAY_OSX_USB];
    [[IDT_BTPay sharedController] setDelegate:self];
}

- (IBAction)close:(id)sender{
    [[IDT_BTPay sharedController] close];
}

- (void) deviceConnected{
    [connectedLabel setStringValue:@"DEVICE CONNECTED"];
}

- (void) deviceDisconnected{
    [connectedLabel setStringValue:@"DEVICE DISCONNECTED"];
}

- (void) dataInOutMonitor:(NSData*)data incoming:(BOOL)isIncoming{
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%s\n%02x", isIncoming?@"IN":@"OUT",
    data.description]];
}

- (void) pinpadData:(NSData*)value keySN:(NSData*)KSN event:(EVENT_PINPAD_Types)event{
    if (event == EVENT_PINPAD_ENCRYPTED_PIN) {

```

```

        [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nPINBLOCK: %@",value.
description]];
        [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nKSN: %@",KSN.description]];
    }
    if (event == EVENT_PINPAD_NUMERIC) {
        [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nNumber: %@", [(NSString alloc)
initWithData:value encoding:NSUTF8StringEncoding] ]];
    }
}

- (void) swipeMSRData:(IDTMSRData*) cardData{
    switch (cardData.event) {
        case EVENT_MSR_CARD_DATA:
        {
            switch (cardData.encryptionType) {
                case CaptureEncodeType_ISOABA:
                    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncryption Type: %
                    @", @"ISO/ABA"]];
                    break;
                case CaptureEncodeType_AAMVA:
                    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncryption Type: %
                    @", @"AA/MVA"]];
                    break;
                case CaptureEncodeType_Other:
                    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncryption Type: %
                    @", @"Other"]];
                    break;
                case CaptureEncodeType_Raw:
                    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncryption Type: %
                    @", @"Raw"]];
                    break;
                default:
                    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncryption Type: %
                    @", @"UNKNOWN"]];
                    break;
            }
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nCard Data: %@", cardData.
cardData]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nTrack 1(%i): %@", cardData
.track1Length, cardData.track1]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nTrack 2(%i): %@",
cardData.track2Length, cardData.track2]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nTrack 3(%i): %@",
cardData.track3Length, cardData.track3]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncoded Track 1: %@",
cardData.encTrack1.description]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncoded Track 2: %@",
cardData.encTrack2.description]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEncoded Track 3: %@",
cardData.encTrack3.description]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nHash Track 1: %@",
cardData.hashTrack1.description]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nHash Track 2: %@",
cardData.hashTrack2.description]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nHash Track 3: %@",
cardData.hashTrack3.description]];
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nKSN: %@", cardData.KSN.
description]];
            return;
        }
        break;

        case EVENT_MSR_CANCEL_KEY:
        {
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%n(Event) MSR Cancel Key rec
eived: %@", cardData.encTrack1]];
            return;
        }
        break;

        case EVENT_MSR_BACKSPACE_KEY:
        {
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%n(Event) MSR Backspace Key
received: %@", cardData.encTrack1]];
            return;
        }
        break;

        case EVENT_MSR_ENTER_KEY:
        {
            [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%n(Event) MSR Enter Key rece
ived: %@", cardData.encTrack1]];
            return;
        }
    }
}

```



```

    }
    break;

    case EVENT_MSR_UNKNOWN:
    {
        [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%n(Event) MSR unknown event,
data: %@", cardData.encTrack1]];
        return;
    }
    break;

    default:
        break;
}

}

-(IBAction) msrON:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController]
    msr_startMSRSwipeWithDisplay:@"Please" line2:@"swipe card"
    line3:nil];
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nEnable MSR Return Status Code %i "
    , rt]];
}

-(IBAction) msrOff:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController]
    msr_cancelMSRSwipe];
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nDisable MSR Return Status Code %i
    ", rt]];
}

-(IBAction) sendUnsecureMessage:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController]
    lcd_displayMessage:@"This is" line2:@"an unsecured" line3:@"message."
    line4:@"Thank you!"];
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nDisplay Message Return Status Code
    %i ", rt]];
}

-(IBAction) device_sendBeep:(id)sender{
    unsigned short beep[] = {0xb00,0x400,0x800,0x300};
    IDT_STATUS rt = [[IDT_BTPay sharedController]
    device_sendBeep:beep numberOfTones:2];
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nControl Beep Return Status Code %i
    ", rt]];
}

-(IBAction) requestPIN:(id)sender{
    IDT_STATUS rt = [[IDT_BTPay sharedController]
    pin_getEncryptedPIN:@"1234567890123456" keyType:PIN_KEY_TDES_DUKPT_extc
    line1:@"Please" line2:@"input PIN" line3:nil];
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nRequest PIN Return Status Code %i
    ", rt]];
}

-(IBAction) requestNumeric:(id)sender{
    NSData* message = [[IDT_BTPay sharedController] getSecureMessage:@"USER ID"];
    IDT_STATUS rt = [[IDT_BTPay sharedController] pin_getNumeric:false minLength:1
    maxLength:16 encryptedDisplayMessage:message];
    [tv.textContainer.textView insertText:[NSString stringWithFormat:@"%nGet Numeric Return Status Code %i
    ", rt]];
}

@end

```

## Chapter 8

# BTPay Error Code Reference

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

---

8900	ATR length too long
8B01	EMV invalid TA1 byte value
8B02	EMV TB1 required
8B03	EMV Unsupported TB1 only 00 allowed
8B04	EMV Card Error, invalid BWI or CWI
8B06	EMV TB2 not allowed in ATR
8B07	EMV TC2 out of range
8B08	EMV TC2 out of range
8B09	per EMV96 TA3 must be > 0xF
8B10	ICC error on power-up
8B11	EMV T=1 then TB3 required
8B12	Card Error, invalid BWI or CWI
8B13	Card Error, invalid BWI or CWI
8B17	EMV TC1/TB3 conflict*
8B20	EMV TD2 out of range must be T=1
8C00	TCK error
A304	connector has no voltage setting
A305	ICC error on power-up invalid (SBLK(IFSD) exchange
E301	ICC error after session start
FF00	Request to go online
FF01	EMV: Accept the offline transaction
FF02	EMV: Decline the offline transaction
FF03	EMV: Accept the online transaction
FF04	EMV: Decline the online transaction
FF05	EMV: Application may fallback to magstripe technology
FF06	EMV: ICC detected tah the conditions of use are not satisfied
FF07	EMV: ICC didn't accept transaction
FF08	EMV: Transaction was cancelled
FF09	EMV: Application was not selected by kernel or ICC format error or ICC missing data error
FF0A	EMV: Transaction is terminated
FF0B	EMV: Other EMV Error
FFFF	NO RESPONSE
0008	err response or data
0009	no reader attached
000A	did connection
000B	mono audio is enabled
000C	audio volume is too low
000D	task or CMD be canceled
0E00	Authorization Accepted
0E01	Unable to go online
0E02	Technical Issue
0E03	Declined
0E04	Issuer Referral transaction
0F00	Accept the online transaction
0F01	Decline the online transaction
0F02	Request to go online
0F03	Transaction is terminated
0F05	Application was not selected by kernel or ICC format error or ICC missing data error
0F07	ICC didn't accept transaction
0F0A	Application may fallback to magstripe technology
0F0C	Transaction was cancelled
0F0D	Timeout
0F0F	Other EMV Error
0F10	Accept the offline transaction
0F11	Decline the offline transaction
0F21	ICC detected tah the conditions of use are not satisfied
0F22	No app were found on card matching terminal configuration
0F23	Terminal file does not exist
0F24	CAPK file does not exist
0F25	CRL Entry does not exist
0FFE	Return code when blocking is disabled
0FFF	Return code when command is not applicable on the selected device

## Chapter 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_UNIPAYII_IOS,
    IDT_DEVICE_UNIPAYII_OSX_USB,
    IDT_DEVICE_IMAG_IOS,
    IDT_DEVICE_VENDI_MOBILE
}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_UNIPAYII_EMV_NO_ICC_MSR_DATA,
    EVENT_UNIPAYII_EMV_FALLBACK_DATA
}EVENT_MSR_Types;
```

```
typedef enum{
    EVENT_ACTIVE_TRANSACTION = 51
}EVENT_CTL5_Types;
```

```
typedef enum {
    RETURN_CODE_DO_SUCCESS = 0,
    RETURN_CODE_ERR_DISCONNECT,
    RETURN_CODE_ERR_CMD_RESPONSE,
    RETURN_CODE_ERR_TIMEDOUT,
    RETURN_CODE_ERR_INVALID_PARAMETER,
    RETURN_CODE_SDK_BUSY_MSR,
    RETURN_CODE_SDK_BUSY_PINPAD,
    RETURN_CODE_SDK_BUSY_CTL5,
    RETURN_CODE_ERR_OTHER,
    RETURN_CODE_FAILED,
    RETURN_CODE_NOT_ATTACHED,
    RETURN_CODE_MONO_AUDIO,
    RETURN_CODE_CONNECTED,
    RETURN_CODE_LOW_VOLUME,
    RETURN_CODE_CANCELED,

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

```

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

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

} RETURN_CODE;

typedef enum{
    EMV_RESULT_CODE_APPROVED = 0X00,
    EMV_RESULT_CODE_DECLINED = 0X01,
    EMV_RESULT_CODE_GO_ONLINE = 0X02,
    EMV_RESULT_CODE_FAILED = 0X03,
    EMV_RESULT_CODE_SYSTEM_ERROR = 0X05,
    EMV_RESULT_CODE_NOT_ACCEPT = 0X07,
    EMV_RESULT_CODE_FALLBACK = 0X0A,
    EMV_RESULT_CODE_CANCEL = 0X0C,
    EMV_RESULT_CODE_OTHER_ERROR = 0X0F,
    EMV_RESULT_CODE_TIME_OUT = 0X0D,
    EMV_RESULT_CODE_OFFLINE_APPROVED = 0X10,
    EMV_RESULT_CODE_OFFLINE_DECLINED = 0X11,
    EMV_RESULT_CODE_REFERRAL_PROCESSING = 0X12,
    EMV_RESULT_CODE_ERROR_APP_PROCESSING = 0X13,
    EMV_RESULT_CODE_ERROR_APP_READING = 0X14,
    EMV_RESULT_CODE_ERROR_DATA_AUTH = 0X15,
    EMV_RESULT_CODE_ERROR_PROCESSING_RESTRICTIONS = 0X16,
    EMV_RESULT_CODE_ERROR_CVM_PROCESSING = 0X17,
    EMV_RESULT_CODE_ERROR_RISK_MGMT = 0X18,
    EMV_RESULT_CODE_ERROR_TERM_ACTION_ANALYSIS = 0X19,
    EMV_RESULT_CODE_ERROR_CARD_ACTION_ANALYSIS = 0X1A,
    EMV_RESULT_CODE_ERROR_APP_SELECTION_TIMEOUT = 0X1B,
    EMV_RESULT_CODE_ERROR_DATA_LEN_INCORRECT = 0X1C,
    EMV_RESULT_CODE_CALL_YOUR_BANK = 0X1D,
    EMV_RESULT_CODE_NO_ICC_ON_CARD = 0X1E,
    EMV_RESULT_CODE_NEW_SELECTION = 0X1F,
    EMV_RESULT_CODE_START_TRANSACTION_SUCCESS = 0X20
} EMV_RESULT_CODE_Types;

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

```

## Chapter 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)
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
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
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)
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
CE	Card Issuer Action Code (PayPass) – Online
CF	Card Issuer Action Code (PayPass) – Decline
D1	Currency conversion table
D2	Integrated Data Storage Directory (IDSD)
D3	Additional check table
D5	Application Control
D6	Default ARPC response code
D7	Application Control (PayPass)
D8	AIP (PayPass)
D9	AFL (PayPass)
DA	Static CVC3-TRACK1
DB	Static CVC3-TRACK2
DC	IVCVC3-TRACK1
DD	IVCVC3-TRACK2
DF01	Encrypted PIN Block in Tag 9F62 – ISO 95641 Format 0
DF02	PEK Version Number
DF03	PIN Try Limit
DF04	PIN Try Counter (VSDC Application)
DF05	AIP - For VISA Contactless
DF06	Products permitted
DF07	Offline checks mandated
DF08	UDKmac
DF09	UDKenc
DF0B	Retries Permitted Limit
DF0C	Script Message Update
DF0D	Fleet Issuer Action Code - Default

DF0E	Fleet Issuer Action Code - Denial
DF0F	Fleet Issuer Action Code - Online
DF12	Vehicle Registration Number
DF13	DDA Public Modulus
DF14	Driver Name
DF15	Driver ID
DF16	Max Fill Volume
DF17	DDA Public Modulus Length
DF18	Mileage
DF20	Issuer Proprietary Bitmap (IPB)
DF21	Internet Authentication Flag (IAF)
DF22	Encrypted PEK - RFU
DF23	PEK Key Check Value - RFU
DF24	MDK - Key derivation Index
DF25	VISA DPA – MDK - Key derivation Index
DF26	Encrypted PIN Block – ISO 9564-1 Format 1 PIN Block (Thales P3 Format 05)
DF40	qVSDC AIP
DF41	VSDC AIP
DF42	UDKac
DF43	UDKmac
DF44	UDKenc
DF47	UDKcvc
DF48	UDKac KCV
DF49	UDKmac KCV
DF4A	UDKenc KCV
DF4B	UDKcvc KCV
DF4B	POS Cardholder Interaction Information
DF51	Grand Parent AC
DF52	Parent AC
DF53	Grand Parent MAC
DF54	Parent MAC
DF55	Grand Parent ENC
DF56	Parent ENC/Terminal Action Code - Default
DF57	Terminal Action Code - Decline
DF60	DS Input (Card)
DF60	DDA Component P
DF61	DDA Component Q
DF61	DS Digest H
DF62	DS ODS Info
DF62	DDA Component D1
DF63	DDA Component D2
DF63	DS ODS Term
DF64	DDA Component Q Minus 1 Mod P
DF65	DDA Private Exponent
DF6B	Paypass Contactless
DF79	Dynamic Data Authentication Keys

DF8101	DS Summary 2
DF8102	DS Summary 3
DF8104	Balance Read Before Gen AC
DF8105	Balance Read After Gen AC
DF8106	Data Needed
DF8107	CDOL1 Related Data
DF8108	DS AC Type
DF8109	DS Input (Term)
DF810A	DS ODS Info For Reader
DF810B	DS Summary Status
DF810C	Kernel ID
DF810D	DSVN Term
DF810E	Post-Gen AC Put Data Status
DF810F	Pre-Gen AC Put Data Status
DF8110	Proceed To First Write Flag
DF8111	PDOL Related Data
DF8112	Tags To Read
DF8113	DRDOL Related Data
DF8114	Reference Control Parameter
DF8115	Error Indication
DF8116	User Interface Request Data
DF8117	Card Data Input Capability
DF8118	CVM Capability – CVM Required
DF8119	CVM Capability – No CVM Required
DF811A	Default UDOL
DF811B	Kernel Configuration
DF811C	Max Lifetime of Torn Transaction Log Record
DF811D	Max Number of Torn Transaction Log Records
DF811E	Mag-stripe CVM Capability – CVM Required
DF811F	Security Capability
DF8120	Terminal Action Code – Default
DF8121	Terminal Action Code – Denial
DF8122	Terminal Action Code – Online
DF8123	Reader Contactless Floor Limit
DF8124	Reader Contactless Transaction Limit (No On-device CVM)
DF8125	Reader Contactless Transaction Limit (On-device CVM)
DF8126	Reader CVM Required Limit
DF8127	Time Out Value
DF8128	IDS Status
DF8129	Outcome Parameter Set
DF812A	DD Card (Track1)
DF812B	DD Card (Track2)
DF812C	Mag-stripe CVM Capability – No CVM Required
DF812D	Message Hold Time
DF8130	Hold Time Value
DF8131	Phone Message Table

---

FF60	Visa International
FF62	Visa Magnetic Stripe
FF63	Visa Quick VSDC
FF8101	Torn Record
FF8102	Tags To Write Before Gen AC
FF8103	Tags To Write After Gen AC
FF8104	Data To Send
FF8105	Data Record
FF8106	Discretionary Data

## Chapter 11

# General Message Table

Secure messages to be used with General Prompts commands

Msg Id	English Prompt	Portuguese Prompt	Spanish Prompt	French Prompt
1	ENTER	ENTER	INGRESE	ENTREZ
2	REENTER	RE-INTRODUZIR	REINGRESE	RE-ENTREZ
3	ENTER YOUR	INTRODUZIR O SEU	INGRESE SU	ENTREZ VOTRE
4	REENTER YOUR	RE-INTRODUZIR O SEU	REINGRESE SU	RE-ENTREZ VOTRE
5	PLEASE ENTER	POR FAVOR DIGITE	POR FAVOR INGRESE	SVP ENTREZ
6	PLEASE REENTER	POR FAVOR REENTRAR	POR FAVOR REINGRESE	SVP RE-ENTREZ
7	PO NUMBER	NÚMERO PO	NUMERO PO	No COMMANDE
8	DRIVER ID	LICENÇA	LICENCIA	ID CONDUCTEUR
9	ODOMETER	ODOMETER	ODOMETRO	ODOMETRE
10	ID NUMBER	NÚMERO ID	NUMERO ID	No IDENT
11	EQUIP CODE	EQUIP CODE	CODIGO EQUIP	CODE EQUIPEMENT
12	DRIVERS ID	DRIVER ID	ID CONDUCTOR	ID CONDUCTEUR
13	JOB NUMBER	EMP NÚMERO	NUMERO EMP	No TRAVAIL
14	WORK ORDER	TRABALHO ORDEM	ORDEN TRABAJO	FICHE TRAVAIL
15	VEHICLE ID	ID VEÍCULO	ID VEHICULO	ID VEHICULE
16	ENTER DRIVER	ENTER DRIVER	INGRESE CONDUCTOR	ENTR CONDUCTEUR
17	ENTER DEPT	ENTER DEPT	INGRESE DEPT	ENTR DEPARTEMNT
18	ENTER PHONE	ADICIONAR PHONE	INGRESE TELEFONO	ENTR No TELEPH
19	ENTER ROUTE	ROUTE ADD	INGRESE RUTA	ENTREZ ROUTE
20	ENTER FLEET	ENTER FROTA	INGRESE FLOTA	ENTREZ PARC AUTO
21	ENTER JOB ID	ENTER JOB ID	INGRESE ID TRABAJO	ENTR ID TRAVAIL



22	ROUTE NUMBER	NÚMERO PATH	RUTA NUMERO	No ROUTE
23	ENTER USER ID	ENTER USER ID	INGRESE ID USUARIO	ID UTILISATEUR
24	FLEET NUMBER	NÚMERO DE FROTA	FLOTA NUMERO	No PARC AUTO
25	ENTER PRODUCT	ADICIONAR PRODUTO	INGRESE PRODUCTO	ENTREZ PRODUIT
26	DRIVER NUMBER	NÚMERO DRIVER	CONDUCTOR NUMERO	No CONDUCTEUR
27	ENTER LICENSE	ENTER LICENÇA	INGRESE LICENCIA	ENTREZ PERMIS
28	ENTER FLEET NO	ENTER NRO FROTA	INGRESE NRO FLOTA	ENT No PARC AUTO
29	ENTER CAR WASH	WASH ENTER	INGRESE LAVADO	ENTREZ LAVE-AUTO
30	ENTER VEHICLE	ENTER VEÍCULO	INGRESE VEHICULO	ENTREZ VEHICULE
31	ENTER TRAILER	TRAILER ENTER	INGRESE TRAILER	ENTREZ REMORQUE
32	ENTER ODOMETER	ENTER ODOMETER	INGRESE ODOMETRO	ENTREZ ODOMETRE
33	DRIVER LICENSE	CARTEIRA DE MOTORISTA	LICENCIA CONDUCTOR	PERMIS CONDUIRE
34	ENTER CUSTOMER	ENTER CLIENTE	INGRESE CLIENTE	ENTREZ CLIENT
35	VEHICLE NUMBER	NÚMERO DO VEÍCULO	VEHICULO NUMERO	No VEHICULE
36	ENTER CUST DATA	ENTER CLIENTE INFO	INGRESE INFO CLIENTE	INFO CLIENT
37	REENTER DRIVID	REENTRAR DRIVER ID	REINGRESE ID CHOFER	RE-ENTR ID COND
38	ENTER USER DATA	ENTER INFO USUÁRIO	INGRESE INFO USUARIO	INFO UTILISATEUR
39	ENTER CUST CODE	ENTER CODE. CLIENTE	INGRESE COD. CLIENTE	ENTR CODE CLIENT
40	ENTER EMPLOYEE	ENTER FUNCIONÁRIO	INGRESE EMPLEADO	ENTREZ EMPLOYE
41	ENTER ID NUMBER	ENTER NÚMERO ID	INGRESE NUMERO ID	ENTREZ No ID
42	ENTER DRIVER ID	ENTER ID DRIVER	INGRESE ID CONDUCTOR	No CONDUCTEUR
43	ENTER FLEET PIN	ENTER PIN FROTA	INGRESE PIN DE FLOTA	NIP PARC AUTO
44	ODOMETER NUMBER	NÚMERO ODOMETER	ODOMETRO NUMERO	No ODOMETRE
45	ENTER DRIVER LIC	ENTER DRIVER LIC	INGRESE LIC CONDUCTOR	PERMIS CONDUIRE
46	ENTER TRAILER NO	NRO TRAILER ENTER	INGRESE NRO TRAILER	ENT No REMORQUE
47	REENTER VEHICLE	REENTRAR VEÍCULO	REINGRESE VEHICULO	RE-ENTR VEHICULE

48	ENTER VEHICLE ID	ENTER VEÍCULO ID	INGRESE ID VEHICULO	ENTR ID VEHICULE
49	ENTER BIRTH DATE	INSERIR DATA NAC	INGRESE FECHA NAC	ENT DT NAISSANCE
50	ENTER DOB MMDDYY	ENTER FDN MMDDYY	INGRESE FDN MMDDAA	NAISSANCE MMJJAA
51	ENTER FLEET DATA	ENTER FROTA INFO	INGRESE INFO DE FLOTA	INFO PARC AUTO
52	ENTER REFERENCE	ENTER REFERÊNCIA	INGRESE REFERENCIA	ENTREZ REFERENCE
53	ENTER AUTH NUMBR	ENTER NÚMERO AUT	INGRESE NUMERO AUT	No AUTORISATION
54	ENTER HUB NUMBER	ENTER HUB NRO	INGRESE NRO HUB	ENTREZ No NOYAU
55	ENTER HUBOMETER	MEDIDA PARA ENTRAR HUB	INGRESE MEDIDO DE HUB	COMPTEUR NOYAU
56	ENTER TRAILER ID	TRAILER ENTER ID	INGRESE ID TRAILER	ENT ID REMORQUE
57	ODOMETER READING	QUILOMETRAGEM	LECTURA ODOMETRO	LECTURE ODOMETRE
58	REENTER ODOMETER	REENTRAR ODOMETER	REINGRESE ODOMETRO	RE-ENT ODOMETRE
59	REENTER DRIV. ID	REENTRAR DRIVER ID	REINGRESE ID CHOFER	RE-ENT ID CONDUC
60	ENTER CUSTOMER ID	ENTER CLIENTE ID	INGRESE ID CLIENTE	ENTREZ ID CLIENT
61	ENTER CUST. ID	ENTER CLIENTE ID	INGRESE ID CLIENTE	ENTREZ ID CLIENT
62	ENTER ROUTE NUM	ENTER NUM ROUTE	INGRESE NUM RUTA	ENT No ROUTE
63	ENTER FLEET NUM	FROTA ENTER NUM	INGRESE NUM FLOTA	ENT No PARC AUTO
64	FLEET PIN	FROTA PIN	PIN DE FLOTA	NIP PARC AUTO
65	DRIVER #	DRIVER #	CONDUCTOR #	CONDUCTEUR
66	ENTER DRIVER #	ENTER DRIVER #	INGRESE CONDUCTOR #	ENT # CONDUCTEUR
67	VEHICLE #	VEÍCULO #	VEHICULO #	# VEHICULE
68	ENTER VEHICLE #	ENTER VEÍCULO #	INGRESE VEHICULO #	ENT # VEHICULE
69	JOB #	TRABALHO #	TRABAJO #	# TRAVAIL
70	ENTER JOB #	ENTER JOB #	INGRESE TRABAJO #	ENTREZ # TRAVAIL
71	DEPT NUMBER	NÚMERO DEPT	NUMERO DEPTO	No DEPARTEMENT
72	DEPARTMENT #	DEPARTAMENTO #	DEPARTAMENTO #	DEPARTEMENT
73	ENTER DEPT #	ENTER DEPT #	INGRESE DEPTO #	ENT# DEPARTEMENT
74	LICENSE NUMBER	NÚMERO DE LICENÇA	NUMERO LICENCIA	No PERMIS
75	LICENSE #	LICENÇA #	LICENCIA #	# PERMIS
76	ENTER LICENSE #	ENTER LICENÇA #	INGRESE LICENCIA #	ENTREZ # PERMIS

77	DATA	INFO	INFO	INFO
78	ENTER DATA	ENTER INFO	INGRESE INFO	ENTREZ INFO
79	CUSTOMER DATA	CLIENTE INFO	INFO CLIENTE	INFO CLIENT
80	ID #	ID #	ID #	# ID
81	ENTER ID #	ENTER ID #	INGRESE ID #	ENTREZ # ID
82	USER ID	USER ID	ID USUARIO	ID UTILISATEUR
83	ROUTE #	ROUTE #	RUTA #	# ROUTE
84	ENTER ROUTE #	ADD ROUTE #	INGRESE RUTA #	ENTREZ # ROUTE
85	ENTER CARD NUM	ENTER NÚMERO DE CARTÃO	INGRESE NUM TARJETA	ENTREZ NO CARTE
86	EXP DATE(Yymm)	VALIDADE VAL (AAMM)	FECHA EXP (AAMM)	DATE EXPIR(AAMM)
87	PHONE NUMBER	TELEFONE	NUMERO TELEFONO	NO TEL
88	CVV START DATE	CVV DATA DE INÍCIO	CVV FECHA INICIO	CVV DATE DE DEBUT
89	ISSUE NUMBER	NÚMERO DE EMISSÃO	NUMERO DE EMISION	NO DEMISSION
90	START DATE (MMYY)	DATA DE INÍCIO (AAMM)	FECHA INICIO (AAMM)	DATE DE DEBUT-AAMM

## Chapter 12

# Hierarchical Index

### 12.1 Class Hierarchy

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

AIDEntry . . . . .	76
ApplicationID . . . . .	78
CAKey . . . . .	80
CRLEntry . . . . .	81
ICCReaderStatus . . . . .	82
<IDT_Device_Delegate>	
IDT_BTPay . . . . .	83
IDT_UniMag . . . . .	125
IDTechEMV . . . . .	133
IDTech_ApplicationID . . . . .	132
IDTech_TerminalData . . . . .	133
MaskAndEncryption . . . . .	144
NSObject	
APDUDData . . . . .	76
APDUResponse . . . . .	77
CTLSResponse . . . . .	81
IDT_BTPay . . . . .	83
IDT_UniMag . . . . .	125
IDTCommon . . . . .	131
IDTechEMV . . . . .	133
IDTEMVData . . . . .	140
IDTMSRData . . . . .	142
UncaughtExceptionHandler . . . . .	147
<NSObject>	
<IDT_BTPay_Delegate> . . . . .	124
<IDT_UniMag_Delegate> . . . . .	130
<IDTechEMV_Delegate> . . . . .	139
PowerOnStructure . . . . .	145
TerminalData . . . . .	146

## Chapter 13

# Data Structure Index

### 13.1 Data Structures

Here are the data structures with brief descriptions:

AIDEntry	76
APDUData	76
APDUResponse	77
ApplicationID	78
CAKey	80
CRLEntry	81
CTLSResponse	81
ICCRReaderStatus	82
IDT_BTPay	83
<IDT_BTPay_Delegate>	124
IDT_UniMag	125
<IDT_UniMag_Delegate>	130
IDTCommon	131
IDTech_ApplicationID	132
IDTech_TerminalData	133
IDTechEMV	133
<IDTechEMV_Delegate>	139
IDTEMVData	140
IDTMSRData	142
MaskAndEncryption	144
PowerOnStructure	145
TerminalData	146
UncaughtExceptionHandler	147

## Chapter 14

# Data Structure Documentation

### 14.1 AIDEntry Struct Reference

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

#### Data Fields

- 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 [emv\\_retrieveAIDList: \(IDT\\_BTPay\)](#) IDT\_UniPay::emv\_retrieveAIDList:().

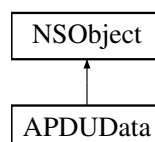
The documentation for this struct was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/IDTCommon.h

### 14.2 APDUData Class Reference

```
#import <APDUData.h>
```

Inheritance diagram for APDUData:



#### Instance Methods

- (void) - [clear](#)

## Class Methods

- (APDUData \*) + sharedController

## Properties

- NSData \* [responseStatus](#)  
*APDU response status bytes, 2 bytes.*
- Byte [maskStatus](#)  
*Mask/Clear status: 0x00=TDDES, 0x10 = AES.*
- BOOL [hasKSN](#)  
*KSN data read.*
- BOOL [hasHash](#)  
*Hash data read.*
- BOOL [hasEncryption](#)  
*APDU response is encrypted.*
- NSData \* [response](#)  
*APDU Response.*
- NSData \* [ksn](#)  
*Key Seral Number.*
- NSData \* [hash](#)  
*Hash value.*

### 14.2.1 Detailed Description

IDT\_UniPay::icc\_exchangeAPDU:response:()

### 14.2.2 Method Documentation

#### 14.2.2.1 - (void) clear

clears all [APDUData](#) properties

#### 14.2.2.2 + (APDUData \*) sharedController

Singleton instance of [APDUData](#)

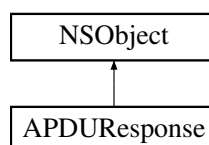
The documentation for this class was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/APDUData.h

## 14.3 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.*

### 14.3.1 Detailed Description

Used in `IDT_BTpay::icc_exchangeAPDU:encrypted:ksn:response:()` `IDT_UniPay::icc_exchangeAPDU:encrypted:response:()`

### 14.3.2 Method Documentation

#### 14.3.2.1 - (void) clear

clears all [APDUResponse](#) properties

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

Singleton instance of [APDUResponse](#)

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

- `/Users/andy/Repo/IDT/IDTech/APDUResponse.h`

## 14.4 ApplicationID Struct Reference

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



## Data Fields

- 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.*

### 14.4.1 Detailed Description

device AID File - 571 bytes

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

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

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

- /Users/randy/Repo/IDT/IDTech/IDTCommon.h

## 14.5 CAKey Struct Reference

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

### Data Fields

- 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.*

### 14.5.1 Detailed Description

Certificate Authority Public Key

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

- /Users/randy/Repo/IDT/IDTech/IDTCommon.h

## 14.6 CRLEntry Struct Reference

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

### Data Fields

- 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.*

### 14.6.1 Detailed Description

Certificate Revocation List Entry - 9 bytes

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

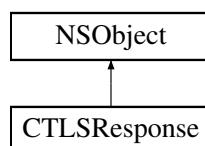
The documentation for this struct was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/IDTCommon.h

## 14.7 CTLSResponse Class Reference

```
#import <CTLSResponse.h>
```

Inheritance diagram for CTLSResponse:



### Instance Methods

- (void) - [clear](#)

### Class Methods

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

### Properties

- NSString \* [track1MSR](#)  
*Track 1 data from MSR.*
- NSString \* [track2MSR](#)  
*Track 2 data from MSR.*

- NSDictionary \* [clearingRecord](#)

*DE 055 data (if available) as a TLV data objects. The DE 055 data is the same data as is included in the Clearing Record.*

- NSDictionary \* [unencryptedTags](#)

*EMV card data.*

- NSDictionary \* [encryptedTags](#)

*EMV card data.*

- NSDictionary \* [maskedTags](#)

*EMV card data.*

### 14.7.1 Detailed Description

Class to encapsulate the [CTLSResponse](#)

### 14.7.2 Method Documentation

#### 14.7.2.1 - (void) clear

clears all [CTLSResponse](#) properties

#### 14.7.2.2 + (CTLSResponse \*) sharedController

Singleton instance of [CTLSResponse](#)

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

- /Users/randy/Repo/IDT/IDTech/CTLSResponse.h

## 14.8 ICCReaderStatus Struct Reference

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

### Data Fields

- bool [iccPower](#)

*Determines if ICC has been powered up.*

- bool [cardSeated](#)

*Determines if card is inserted.*

- bool [latchClosed](#)

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

- bool [cardPresent](#)

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

- bool [magneticDataPresent](#)

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

### 14.8.1 Detailed Description

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

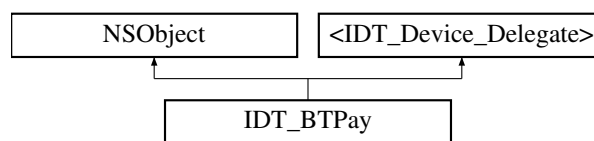
The documentation for this struct was generated from the following file:

- /Users/andy/Repo/IDT/IDTech/IDTCommon.h

## 14.9 IDT\_BTPay Class Reference

```
#import <IDT_BTPay.h>
```

Inheritance diagram for IDT\_BTPay:



### Instance Methods

- (RETURN\_CODE) - [config\\_getDateTime:](#)
- (RETURN\_CODE) - [config\\_getInterfaceDeviceSN:](#)
- (RETURN\_CODE) - [config\\_getLanguageType:](#)
- (RETURN\_CODE) - [device\\_getMaskAndEncryption:](#)
- (RETURN\_CODE) - [config\\_getModelNumber:](#)
- (RETURN\_CODE) - [config\\_getSerialNumber:](#)
- (RETURN\_CODE) - [config\\_getTerminalIdentification:](#)
- (RETURN\_CODE) - [config\\_setBluetoothAddress:](#)
- (BOOL) - [config\\_setCmdTimeOutDuration:](#)
- (RETURN\_CODE) - [config\\_setDateTime:](#)
- (RETURN\_CODE) - [config\\_setInterfaceDeviceSN:](#)
- (RETURN\_CODE) - [config\\_setLanguageType:](#)
- (RETURN\_CODE) - [config\\_setSerialNumber:](#)
- (RETURN\_CODE) - [config\\_setTerminalIdentification:](#)
- (void) - [config\\_stayConnected:](#)
- (RETURN\_CODE) - [device\\_getFirmwareVersion:](#)
- (NSString \*) - [device\\_getResponseCodeString:](#)
- (bool) - [device\\_isConnected:](#)
- (RETURN\_CODE) - [device\\_sendBeep:numberOfTones:](#)
- (RETURN\_CODE) - [device\\_sendDataCommand:calcLRC:response:](#)
- (RETURN\_CODE) - [device\\_restoreMaskAndEncryptionDefaults](#)
- (RETURN\_CODE) - [device\\_setExpMasking:](#)
- (RETURN\_CODE) - [device\\_setPANMaskingCharacter:](#)
- (RETURN\_CODE) - [device\\_setPostPANClearDigits:](#)
- (RETURN\_CODE) - [device\\_setPrePANClearDigits:](#)
- (RETURN\_CODE) - [device\\_rebootDevice](#)
- (RETURN\_CODE) - [device\\_uploadJPEG:](#)
- (RETURN\_CODE) - [emv\\_completeOnlineEMVTransaction:hostResponseTags:responseTags:](#)
- (RETURN\_CODE) - [emv\\_getEMVKernelVersion:](#)
- (RETURN\_CODE) - [emv\\_getAllTags:](#)
- (RETURN\_CODE) - [emv\\_getTag:tagData:](#)

- (RETURN\_CODE) - [emv\\_removeApplicationData:](#)
- (RETURN\_CODE) - [emv\\_removeCAPK:index:](#)
- (RETURN\_CODE) - [emv\\_removeCRL:](#)
- (RETURN\_CODE) - [emv\\_removeCRLUnit:](#)
- (RETURN\_CODE) - [emv\\_removeTerminalData](#)
- (RETURN\_CODE) - [emv\\_retrieveAIDList:](#)
- (RETURN\_CODE) - [emv\\_retrieveApplicationData:response:](#)
- (RETURN\_CODE) - [emv\\_retrieveCAPK:index:response:](#)
- (RETURN\_CODE) - [emv\\_retrieveCAPKList:](#)
- (RETURN\_CODE) - [emv\\_retrieveCRLForRID:response:](#)
- (RETURN\_CODE) - [emv\\_retrieveCRLList:](#)
- (RETURN\_CODE) - [emv\\_retrieveTerminalData:](#)
- (RETURN\_CODE) - [emv\\_setApplicationData:](#)
- (RETURN\_CODE) - [emv\\_setCAPK:](#)
- (RETURN\_CODE) - [emv\\_setCRL:](#)
- (RETURN\_CODE) - [emv\\_setTerminalData:](#)
- (RETURN\_CODE) - [emv\\_startEMVTransaction:otherAmount:timeout:transactionType:additionalTags:](#)
- (RETURN\_CODE) - [icc\\_exchangeAPDU:response:](#)
- (RETURN\_CODE) - [icc\\_exchangeEncryptedAPDU:ksn:response:](#)
- (RETURN\_CODE) - [icc\\_getAPDU\\_KSN:](#)
- (RETURN\_CODE) - [icc\\_getICCRReaderStatus:](#)
- (RETURN\_CODE) - [icc\\_powerOffICC:](#)
- (RETURN\_CODE) - [icc\\_powerOnICC:](#)
- (RETURN\_CODE) - [icc\\_powerOnICC:response:](#)
- (RETURN\_CODE) - [lcd\\_showJPEG:Y0:X1:Y1:](#)
- (RETURN\_CODE) - [lcd\\_displayMessage:line2:line3:line4:](#)
- (RETURN\_CODE) - [lcd\\_enterSleepMode](#)
- (RETURN\_CODE) - [lcd\\_setIdleTimeForSleep:](#)
- (RETURN\_CODE) - [msr\\_cancelMSRSwipe](#)
- (RETURN\_CODE) - [msr\\_getCardDataEncryptedAlgorithm:](#)
- (RETURN\_CODE) - [msr\\_getEncryptMSRFormat:](#)
- (RETURN\_CODE) - [msr\\_setCardDataEncryptedAlgorithm:](#)
- (RETURN\_CODE) - [msr\\_setEncryptMSRFormat:](#)
- (RETURN\_CODE) - [msr\\_startMSRSwipeWithDisplay:line2:line3:](#)
- (RETURN\_CODE) - [pin\\_cancelPin](#)
- (RETURN\_CODE) - [pin\\_getAmount:maxLength:messageID:language:](#)
- (RETURN\_CODE) - [pin\\_getCardAccount:max:line1:line2:](#)
- (RETURN\_CODE) - [pin\\_getFunctionKey](#)
- (RETURN\_CODE) - [pin\\_getPinLength:](#)
- (RETURN\_CODE) - [pin\\_getPINpadStatus:](#)
- (RETURN\_CODE) - [pin\\_getEncryptedData:minLength:maxLength:messageID:language:](#)
- (RETURN\_CODE) - [pin\\_getEncryptedPIN:keyType:line1:line2:line3:](#)
- (RETURN\_CODE) - [pin\\_getNumeric:minLength:maxLength:messageID:language:](#)
- (RETURN\_CODE) - [pin\\_setPinLength:maxLength:](#)
- (void) - [close](#)
- (bool) - [isConnected](#)
- (void) - [attemptConnect](#)

## Class Methods

- (NSString \*) + [SDK\\_version](#)
- (IDT\_BTPay \*) + [sharedController](#)
- (void) + [connectUSB:](#)

## Properties

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

### 14.9.1 Detailed Description

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

### 14.9.2 Method Documentation

#### 14.9.2.1 - (void) attemptConnect

Attempt connection

Requests a connection attempt

#### 14.9.2.2 - (void) close

Close Device

#### 14.9.2.3 - (RETURN\_CODE) config\_getDateTime: (NSString \*\*) response

Polls device for current Date/Time

##### Parameters

<i>response</i>	Response returned as ASCII Data of Date YYMMDDhhmmss. Example 140215171628 = Feb. 15, 2014, 28 seconds into 5:16pm.
-----------------	---

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

```
NSString* response;
RETURN_CODE rt = [[IDT_BTPay_sharedController] config_getDateTime:&response];
if (RETURN_CODE_DO_SUCCESS == rt)
{
    LOGI* ("%Date Time (YYMMDDhhmmss) = %@", response);
}
```

#### 14.9.2.4 - (RETURN\_CODE) config\_getInterfaceDeviceSN: (NSString \*\*) response

Get interface device's serial number

##### Parameters

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

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.5 - (RETURN\_CODE) config\_getLanguageType: (NSUInteger \*\*) response

Get Language Type

Gets the language type



## Parameters

<i>response</i>	LANGUAGE_TYPE of the BTPay ENGLISH: 01 PORTUGUESE: 02 SPANISH: 03
-----------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.6 - (RETURN\_CODE) config\_getModelNumber: (NSString \*\*) *response*

Polls device for Model Number

## Parameters

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

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

Polls device for Serial Number

## Parameters

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.8 - (RETURN\_CODE) config\_getTerminalIdentification: (NSString \*\*) *response*

Get terminal identification

## Parameters

<i>response</i>	Returns device terminal identification
-----------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.9 - (RETURN\_CODE) config\_setBluetoothAddress: (NSString \*) *address*

Set Bluetooth Address

Sets the Bluetooth address of the device. 6 bytes, example F0DE07CCA03F.

## Parameters

<i>address</i>	6 Byte address represented by a 12-character HEX string
----------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.10 - (BOOL) config\_setCmdTimeOutDuration: (int) *nSecond*

## Command Acknowledgement Timeout

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

## Parameters

<i>nSecond</i>	Timeout value. Valid range 1 - 60 seconds
----------------	---

## Return values

<i>Success</i>	flag. Determines if value was set and in range.
----------------	---

14.9.2.11 - (RETURN\_CODE) config\_setDateTime: (NSString \*) *date*

## Set device Date/Time

Set device's date/time

## Parameters

<i>date</i>	Device date represented by a YYMMDDhhmmss. Example March 12, 2014, 6:30pm (and 12 seconds) = 140312183012
-------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.12 - (RETURN\_CODE) config\_setInterfaceDeviceSN: (NSString \*) *sn*

Set Interface Device serial number.

EMV serial number can be set only once

## Parameters

<i>sn</i>	Device serial number
-----------	----------------------

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.13 - (RETURN\_CODE) config\_setLanguageType: (LANGUAGE\_TYPE) *lang*

## Set Language Type

Sets the language type of BTPay prompts

## Parameters

<i>lang</i>	LANGUAGE_TYPE ENGLISH: 01 PORTUGUESE: 02 SPANISH: 03
-------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> .)
--------------------	--

14.9.2.14 - (RETURN\_CODE) config\_setSerialNumber: (NSString \*) *strSN*

## Set Serial Number

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

## Parameters

<i>strSN</i>	Device serial number must be 10 characters
--------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.15 - (RETURN\_CODE) config\_setTerminalIdentification: (NSString \*) *sn*

## Set terminal identification

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

## Parameters

<i>sn</i>	Device terminal identification
-----------	--------------------------------

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.16 - (void) config\_stayConnected: (BOOL) *stayConnected*

## Keep Accessory Connected

Framework attempts to disconnect from accessory whenever application goes to background. Setting this value to TRUE will disable framework disconnect attempts. NOTE: ExternalAccessory may still disconnect device when going to background by default. If you want to stay connected to device, you must also set the .plist "Required Background Modes" to "App communicates using CoreBluetooth", "App communicates with an accessory", and "App shares data using CoreBluetooth"

## Parameters

<i>stayConnected</i>	TRUE = stay connected while in background (assuming .plist is properly configured)
----------------------	--

14.9.2.17 + (void) connectUSB: (BOOL) *isUSB*

## Sets the OSX Connection Method

When using BTPay on OSX, the device can connect either via Bluetooth or USB-HID. Default is Bluetooth. Use this function to change the connection method to USB-HID

#### Parameters

<i>isUSB</i>	TRUE = connect via USB-HID, FALSE = connect via BlueTooth.
--------------	--

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

Polls device for Firmware Version

#### Parameters

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

#### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.19 - (RETURN\_CODE) device\_getMaskAndEncryption: (MaskAndEncryption \*\*) data

Get Mask and Encryption

Retrieves the MSR Mask and Encryption settings

#### Parameters

<i>data</i>	Pointer that will return location of <a href="#">MaskAndEncryption</a> structure.
-------------	---

#### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.20 - (NSString \*) device\_getResponseCodeString: (int) errorCode

Get Response Code String

- All Devices

Interpret a [IDT\\_BTPay](#) response code and return string description.

#### Parameters

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

#### Return values

<i>Verbose</i>	error description
----------------	-------------------

HEX VALUE	Description
0x0000	No error, beginning task
0x0001	No response from reader
0x0002	Invalid response data
0x0003	Time out for task or CMD

0x0004	Wrong parameter
0x0005	SDK is doing MSR or ICC task
0x0006	SDK is doing PINPad task
0x0007	SDK is doing Other task
0x0300	Key Type(TDES) of Session Key is not same as the related Master Key.
0x0400	Related Key was not loaded.
0x0500	Key Same.
0x0702	PAN is Error Key.
0x0D00	This Key had been loaded.
0x0E00	Base Time was loaded.
0x1800	Send "Cancel Command" after send "Get Encrypted PIN" & "Get Numeric" & "Get Amount"
0x1900	Press "Cancel" key after send "Get Encrypted PIN" & "Get Numeric" & "Get Amount"
0x30FF	Security Chip is not connect
0x3000	Security Chip is deactivation & Device is In Removal Legally State.
0x3101	Security Chip is activation & Device is In Removal Legally State.
0x5500	No Admin DUKPT Key.
0x5501	Admin DUKPT Key STOP.
0x5502	Admin DUKPT Key KSN is Error.
0x5503	Get Authentication Code1 Failed.
0x5504	Validate Authentication Code Error.
0x5505	Encrypt or Decrypt data failed.
0x5506	Not Support the New Key Type.
0x5507	New Key Index is Error.
0x5508	Step Error.
0x550F	Other Error.
0x6000	Save or Config Failed / Or Read Config Error.
0x6200	No Serial Number.
0x6900	Invalid Command - Protocol is right, but task ID is invalid.
0x6A00	Unsupported Command - Protocol and task ID are right, but command is invalid.
0x6B00	Unknown parameter in command - Protocol task ID and command are right, but parameter is invalid.
0x7200	Device is suspend (MKSK suspend or press password suspend).
0x7300	PIN DUKPT is STOP (21 bit 1).
0x7400	Device is Busy.
0xE100	Can not enter sleep mode.
0xE200	File has existed.
0xE300	File has not existed.
0xE400	Open File Error.
0xE500	SmartCard Error.
0xE600	Get MSR Card data is error.
0xE700	Command time out.

0xE800	File read or write is error.
0xE900	Active 1850 error!
0xEA00	Load bootloader error.
0xEF00	Protocol Error- STX or ETX or check error.
0xEB00	Picture is not exist.
0x2C06	no card seated to request ATR
0x2D01	Card Not Supported,
0x2D03	Card Not Supported, wants CRC
0x690D	Command not supported on reader without ICC support
0x8100	ICC error time out on power-up
0x8200	invalid TS character received
0x8500	pps confirmation error
0x8600	Unsupported F, D, or combination of F and D
0x8700	protocol not supported EMV TD1 out of range
0x8800	power not at proper level
0x8900	ATR length too long
0x8B01	EMV invalid TA1 byte value
0x8B02	EMV TB1 required
0x8B03	EMV Unsupported TB1 only 00 allowed
0x8B04	EMV Card Error, invalid BWI or CWI
0x8B06	EMV TB2 not allowed in ATR
0x8B07	EMV TC2 out of range
0x8B08	EMV TC2 out of range
0x8B09	per EMV96 TA3 must be > 0xF
0x8B10	ICC error on power-up
0x8B11	EMV T=1 then TB3 required
0x8B12	Card Error, invalid BWI or CWI
0x8B13	Card Error, invalid BWI or CWI
0x8B17	EMV TC1/TB3 conflict*
0x8B20	EMV TD2 out of range must be T=1
0x8C00	TCK error
0xA304	connector has no voltage setting
0xA305	ICC error on power-up invalid (SBLK(IFSD) exchange
0xE301	ICC error after session star
0xFF00	EMV: Request to go online
0xFF01	EMV: Accept the offline transaction
0xFF02	EMV: Decline the offline transaction
0xFF03	EMV: Accept the online transaction
0xFF04	EMV: Decline the online transaction
0xFF05	EMV: Application may fallback to magstripe technology
0xFF06	EMV: ICC detected that the conditions of use are not satisfied
0xFF07	EMV: ICC didn't accept transaction
0xFF08	EMV: Transaction was cancelled
0xFF09	EMV: Application was not selected by kernel or ICC format error or ICC missing data error

0xFF0A	EMV: Transaction is terminated
0xFF0B	EMV: Other EMV Error

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

##### Is Device Connected

Returns the connection status of the BTPay

##### Parameters

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

```
typedef enum{
    IDT_DEVICE_BTPay_IOS = 0,
    IDT_DEVICE_BTPay_OSX_BT,
    IDT_DEVICE_BTPay_OSX_USB
} IDT_DEVICE_Types;
```

#### 14.9.2.22 - (RETURN\_CODE) device\_rebootDevice

##### Reboot Device

Executes a command to restart the device.

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.23 - (RETURN\_CODE) device\_restoreMaskAndEncryptionDefaults

Restore Mask and Encryption default settings

- BTPay

Restores the default values for MSR Mask and Encryption settings

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.24 - (RETURN\_CODE) device\_sendBeep: (unsigned short \*) beep numberOfTones:(int) num

Sends a Beep Value

Executes a beep on the BT200. The complete beep may be defined as a multiple of single beep tones.

##### Parameters

<i>beep</i>	Unsigned short array containing freq1,dur1,freq2,dur2,. . . freq#,dur#. Frequency is in Hz and must be in the range 2000-4000. Duration is in milliseconds.
<i>num</i>	Number of tones in the beep array.

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## Example Code-

```
unsigned short beep[] = {0xb00,0x400,0x800,0x300};
RETURN_CODE rt = [[IDT_BTPay sharedController] device_sendBeep:beep numberOfTones:
    2];
LOGI(@"\nControl Beep Return Status Code %i ", rts);
```

14.9.2.25 - (RETURN\_CODE) device\_sendDataCommand: (NSData \*) *cmd* calcLRC:(BOOL) *lrc* response:(NSData \*\*) *response*

Send a NSData object to device

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

## Parameters

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.26 - (RETURN\_CODE) device\_setExpMasking: (BOOL) *mask*

Set Expiration Date masking

- BTPay

Sets the flag to enable Expiratin Date masking

## Parameters

<i>mask</i>	TRUE = mask expiration date. FALSE = display expiration date in cleartext
-------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.27 - (RETURN\_CODE) device\_setPANMaskingCharacter: (char) *maskChar*

Set PAN masking character

- BTPay

Sets the character for PAN masking



## Parameters

<i>maskChar</i>	Masking character. Default value '*';
-----------------	---------------------------------------

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> ;
--------------------	---

14.9.2.28 - (RETURN\_CODE) device\_setPostPANClearDigits: (int) *clearDigits*

## Set PostPAN Clear Digits

- BTPay

Sets the number of digits to show in clear text at the ending of PAN

## Parameters

<i>clearDigits</i>	Amount of characters to display cleartext at end of PAN. Valid range 0-4. Default value 4.
--------------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> ;
--------------------	---

14.9.2.29 - (RETURN\_CODE) device\_setPrePANClearDigits: (int) *clearDigits*

## Set PrePAN Clear Digits

- BTPay

Sets the number of digits to show in clear text at the beginning of PAN

## Parameters

<i>clearDigits</i>	Amount of characters to display cleartext at beginning of PAN. Valid range 0-6. Default value 4.
--------------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> ;
--------------------	---

14.9.2.30 - (RETURN\_CODE) device\_uploadJPEG: (NSData \*) *picture*

## Upload JPEG to device

Stores a picture on the device. The picture's dimensions must not exceed the display resolution of 128 x 64. The picture must be RGB JPEG.

## Parameters

<i>picture</i>	RGB JPEG image data
----------------	---------------------

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.31 - (RETURN\_CODE) emv\_completeOnlineEMVTransaction: (EMV\_AUTHORIZATION\_RESULT) *result*  
 hostResponseTags:(NSMutableDictionary \*) *tags* responseTags:(IDTEMVData \*\*) *response*

## 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>result</i>	Determines if connection to host was successful.
<i>tags</i>	Host response tag
<i>response</i>	returns the response tags

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## Converting TLV to NSMutableDictionary

EMV data is received in TLV (Tag, Length, value) format: 950500000080009B02E8009F2701018A025A339F26080C552B9364D55CE5

This data contains the following EMV tags/values:

Tag	Length	Value
95	05	0000008000
9B	02	E800
9F27	01	01
8A	02	5A33
9F26	08	0C552B9364D55CE5

An example how to create an NSMutableDictionary with these values follows.

```
-(NSMutableDictionary*) createTLVDict{
NSMutableDictionary *emvTags = [[NSMutableDictionary alloc] initWithCapacity:0];

[emvTags setObject:@"0000008000" forKey:@"95"];
[emvTags setObject:@"E800" forKey:@"9B"];
[emvTags setObject:@"01" forKey:@"9F27"];
[emvTags setObject:@"5A33" forKey:@"8A"];
[emvTags setObject:@"0C552B9364D55CE5" forKey:@"9F26"];

return emvTags;
}
```

14.9.2.32 - (RETURN\_CODE) emv\_getAllTags: (NSDictionary \*\*) *data*

## Get All Tags

Retrieves all EMV tags from the inserted card. Only available after the card has been processed after executing [emv\\_startEMVTransaction:otherAmount:timeout:transactionType:additionalTags:](#)

## Parameters

<i>data</i>	Pointer that will return location of dictionary with all tag values. Key is NSString, Object is NSData.
-------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString:</a>
--------------------	--

14.9.2.33 - (RETURN\_CODE) emv\_getEMVKernelVersion: (NSString \*\*) *response*

## Polls device for EMV Kernel Version

## Parameters

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString:</a>
--------------------	--

14.9.2.34 - (RETURN\_CODE) emv\_getTag: (NSString \*) *tagName* tagData:(NSData \*\*) *data*

## Get Tag

Retrieves an EMV tag from the inserted card. Only available after the card has been processed after executing [emv\\_startEMVTransaction:otherAmount:timeout:transactionType:additionalTags:](#)

## Parameters

<i>tagName</i>	Name fo tag to retrieve
<i>data</i>	Pointer that will return location of tag data

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString:</a>
--------------------	--

14.9.2.35 - (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
------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.36 - (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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.37 - (RETURN\_CODE) emv\_removeCRL: (CRLEntry) *key*

## Remove Certificate Revocation List RID

Removes all [CRLEntry](#) as specified by the RID and Index passed as a parameter in the [CRLEntry](#) structure

## Parameters

<i>key</i>	<a href="#">CRLEntry</a> containing the RID and Index of the of the entries to remove
------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.38 - (RETURN\_CODE) emv\_removeCRLUnit: (CRLEntry) *key*

## Remove Certificate Revocation List unit

Removes a single [CRLEntry](#) as specified by the RID/Index/Serial Number passed as a parameter in the [CRLEntry](#) structure

## Parameters

<i>key</i>	<a href="#">CRLEntry</a> containing the RID, Index and serial number of the of the entry to remove
------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

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

## Remove Terminal Data

Removes the Terminal Data

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

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

## Retrieve AID list

Returns all the AID name/length on the inserted ICC. Populates response parameter with an dictionary with Keys of AID Names (NSData\*), and values of AID Lengths (NSData\*)

## Parameters

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.41 - (RETURN\_CODE) emv\_retrieveApplicationData: (NSString \*) AID response:(ApplicationID \*\*) responseAID

## Retrieve Application Data by AID

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.42 - (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>

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.43 - (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
-----------------	--

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.44 - (RETURN\_CODE) emv\_retrieveCRLForRID: (NSString \*) rid response:(NSArray \*\*) response

Retrieve the Certificate Revocation List specific to RID and index

Returns all the serial numbers for a specific RID/Index on the CRL.

##### Parameters

<i>rid</i>	RID of the certificate to search for
<i>response</i>	Response returned as an NSArray of NSData* objects for each <a href="#">CRLEntry</a> : 5 bytes: AID 1 byte: Index 3 bytes: Serial Number

The following code can map the NSData entries into crlEntries

```
NSArray* returnArray;
[[IDT_BTPay_sharedController] emv_retrieveCRLForRID:@"a000000003" response:&
returnArray];
for (NSData* obj in returnArray) {
Byte *keyByte = (Byte*) obj.bytes;
CRLEntry* crl = (CRLEntry*)keyByte;
}
```

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.45 - (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 5-byte objects for each RID
-----------------	---

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.46 - (RETURN\_CODE) emv\_retrieveTerminalData: (TerminalData \*\*) response

Retrieve Terminal Data

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

## Parameters

<i>response</i>	Response returned as a <a href="#">TerminalData</a>
-----------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.47 - (RETURN\_CODE) emv\_setApplicationData: ([ApplicationID](#)) *data*

## Set Application Data by AID

Sets the Application Data as specified by the [ApplicationID](#) structure passed as a parameter

## Parameters

<i>data</i>	<a href="#">ApplicationID</a> configuration file
-------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.48 - (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
------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.49 - (RETURN\_CODE) emv\_setCRL: ([CRLEntry](#)) *key*

## Set Certificate Revocation List Entry

Sets the CRL entry as specified by the [CRLEntry](#) structure

## Parameters

<i>key</i>	<a href="#">CRLEntry</a> containing the RID, Index, and serial number to set
------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.50 - (RETURN\_CODE) emv\_setTerminalData: ([TerminalData](#)) *data*

## Set Terminal Data

Sets the Terminal Data as specified by the [TerminalData](#) structure passed as a parameter



## Parameters

<i>data</i>	<a href="#">TerminalData</a> configuration file
-------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.51 - (RETURN\_CODE) emv\_startEMVTransaction: (float) *amount* otherAmount:(float) *amtOther* timeout:(int) *timeout* transactionType:(unsigned char) *type* additionalTags:(NSMutableDictionary \*) *tags*

Start EMV Transaction Request

Authorizes the EMV transaction for an ICC card

The tags will be returned in the emvTransactionData delegate protocol.

## Parameters

<i>amount</i>	Transaction amount value (tag value 9F02)
<i>amtOther</i>	Other amount value, if any (tag value 9F03)
<i>timeout</i>	Timeout value in seconds.
<i>type</i>	Transaction type (tag value 9C).
<i>tags</i>	Any other optional tags to be included in the request. Passed as a mutable dictionary.

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## Converting TLV to NSMutableDictionary

EMV data is received in TLV (Tag, Length, value) format: 950500000080009B02E8009F2701018A025A339F26080C552B9364D55CE5

This data contains the following EMV tags/values:

Tag	Length	Value
95	05	0000008000
9B	02	E800
9F27	01	01
8A	02	5A33
9F26	08	0C552B9364D55CE5

An example how to create an NSMutableDictionary with these values follows.

```
-(NSMutableDictionary*) createTLVDict{
    NSMutableDictionary *emvTags = [[NSMutableDictionary alloc] initWithCapacity:0];

    [emvTags setObject:@"0000008000" forKey:@"95"];
    [emvTags setObject:@"E800" forKey:@"9B"];
    [emvTags setObject:@"01" forKey:@"9F27"];
    [emvTags setObject:@"5A33" forKey:@"8A"];
    [emvTags setObject:@"0C552B9364D55CE5" forKey:@"9F26"];

    return emvTags;
}
```

#### 14.9.2.52 - (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

Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.53 - (RETURN\_CODE) `icc_exchangeEncryptedAPDU: (NSData *) dataAPDU ksn:(NSData *) ksn response:(APDUResponse **) response`

Exchange Encrypted APDU

Sends an Encrypted 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>ksn</i>	KSN Value
<i>response</i>	Unencrypted/encrypted parsed APDU response

Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### 14.9.2.54 - (RETURN\_CODE) `icc_getAPDU_KSN: (NSData **) ksn`

Get APDU KSN

Retrieves the KSN used in ICC Encrypted APDU usage

Parameters

<i>ksn</i>	Returns the encrypted APDU packet KSN
------------	---------------------------------------

Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a>
--------------------	--

#### 14.9.2.55 - (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="#">ICCRedReaderStatus</a> results.
---------------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

```

ICCRedReaderStatus readerStatus;
RETURN_CODE rt = [[IDT_BTPay sharedController] icc_getICCRedReaderStatus:&
    readerStatus];
if (RETURN_CODE_DO_SUCCESS != rt) {
    LOGI(@"Fail");
}
else {
    NSString *sta;
    if (readerStatus.iccPower)
        sta = @"[ICC Powered]";
    else
        sta = @"[ICC Power not Ready]";
    if (readerStatus.cardSeated)
        sta = [NSString stringWithFormat:@"%0", [Card Seated]", sta];
    else
        sta = [NSString stringWithFormat:@"%0", [Card not Seated]", sta];
    LOGI(@"Card Status = %0", sta);
}

```

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

## Power Off ICC

Powers down the ICC

## Parameters

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

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

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

14.9.2.57 - (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
-----------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.58 - (RETURN\_CODE) icc\_powerOnICC: (PowerOnStructure) *options* response:(NSData \*\*) *response*

## Power On ICC with Options

Power up the currently selected microprocessor card in the ICC reader, specifying IFS/pps options.

## Parameters

<i>options</i>	ATR Options
<i>response</i>	Response returned. If Success, ATR String. If Failure, ASCII encoded data of error string

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.59 - (bool) isConnected

Check if device is connected

14.9.2.60 - (RETURN\_CODE) lcd\_displayMessage: (NSString \*) *line1* line2:(NSString \*) *line2* line3:(NSString \*) *line3* line4:(NSString \*) *line4*

DisplayMessage

Display up to 4 lines of text in the device LCD.

## Parameters

<i>line1</i>	Display line 1, up to 12 characters
<i>line2</i>	Display line 2, up to 16 characters
<i>line3</i>	Display line 3, up to 16 characters
<i>line4</i>	Display line 4, up to 16 characters

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.61 - (RETURN\_CODE) lcd\_enterSleepMode

Put device to sleep

Set device to enter sleep mode. In sleep mode, LCD display and backlight is off. It can be waked up by key press or sending commands

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.62 - (RETURN\_CODE) lcd\_setIdleTimeForSleep: (int) *seconds*

Set Enter Sleep Mode Time

Sets seconds of idle that must pass before entering sleep mode

## Parameters

<i>seconds</i>	Amount of time (in seconds) that must pass during idle before unit goes to sleep
----------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.63 - (RETURN\_CODE) lcd\_showJPEG: (int) X0 Y0:(int) Y0 X1:(int) X1 Y1:(int) Y1

Show stored picture on the LCD

Show stored picture on the LCD defined by top left point [X0][Y0] and bottom right point [X1][Y1]. The values of X must be in the range 0-127, and the values of Y must be in the range of 0-63

## Parameters

<i>X0</i>	Upper left X coordinate
<i>Y0</i>	Upper left Y coordinate
<i>X1</i>	Lower left X coordinate
<i>Y1</i>	Lower left Y coordinate

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.64 - (RETURN\_CODE) msr\_cancelMSRSwipe

Disable MSR Swipe

Cancels MSR swipe request.

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.65 - (RETURN\_CODE) msr\_getCardDataEncryptedAlgorithm: (NSString \*\*) response

Get Card Data Encrypted Algorithm

Sets the encrypted algorithm of MSR card data and SmartCard data (if MSR DUKPT key loaded

## Parameters

<i>response</i>	Response returned from method: <ul style="list-style-type: none"> <li>• '0': Encrypted card data with Data Key if MSR DUKPT Key had been loaded.(default)</li> <li>• '1': Encrypted card data with PIN Key if MSR DUKPT Key had been loaded.</li> </ul>
-----------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.66 - (RETURN\_CODE) msr\_getEncryptMSRFormat: (NSString \*\*) response

Get encrypted MSR Data Output Format

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

## Parameters

<i>response</i>	Response returned from method: <ul style="list-style-type: none"> <li>• '0': Encrypted card data with Data Key if MSR DUKPT Key had been loaded.(default)</li> <li>• '1': Encrypted card data with PIN Key if MSR DUKPT Key had been loaded.</li> </ul>
-----------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.67 - (RETURN\_CODE) msr\_setCardDataEncryptedAlgorithm: (int) *encryption*

## Set Card Data Encrypted Algorithm

Sets the encrypted algorithm of MSR card data and SmartCard data (if MSR DUKPT key loaded)

## Parameters

<i>encryption</i>	Encryption Type <ul style="list-style-type: none"> <li>• 01: 3DES (Default)</li> <li>• 02: AES</li> </ul>
-------------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.68 - (RETURN\_CODE) msr\_setEncryptMSRFormat: (int) *encryption*

## Set encrypted MSR Data Output Format

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

## Parameters

<i>encryption</i>	Encryption Type <ul style="list-style-type: none"> <li>• 00: Encrypt with Data Key</li> <li>• 01: Encrypt with PIN Key</li> </ul>
-------------------	---

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.9.2.69 - (RETURN\_CODE) msr\_startMSRSwipeWithDisplay: (NSString \*) *line1* line2:(NSString \*) *line2* line3:(NSString \*) *line3*

## Display Message and Enable MSR Swipe

Prompts the user with up to 3 lines of text. Enables MSR, waiting for swipe to occur. Returns [IDTMSRData](#) instance to `deviceDelegate::swipeMSRData:()`

During waiting for swiping card, it will receive all commands except [pin\\_getEncryptedPIN:keyType:line1:line2:line3:](#) and [pin\\_getNumeric:minLength:maxLength:messageID:language:](#) and [pin\\_getAmount:maxLength:messageID:language:](#) and [pin\\_getCardAccount:max:line1:line2:](#) and [pin\\_getEncryptedData:minLength:maxLength:messageID:language:](#)

#### Parameters

<i>line1</i>	Display line 1, up to 12 characters
<i>line2</i>	Display line 2, up to 16 characters
<i>line3</i>	Display line 3, up to 16 characters
	•

#### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString:</a>
--------------------	--

#### 14.9.2.70 - (RETURN\_CODE) pin\_cancelPin

##### Cancel PIN Command

This command can cancel [IDT\\_BTPay:pin\\_getEncryptedPIN:keyType:line1:line2:line3:\(\)](#) and [pin\\_getNumeric:minLength:maxLength:messageID:language:](#) and [pin\\_getAmount:maxLength:messageID:language:](#) and [pin\\_getCardAccount:max:line1:line2:](#) and [pin\\_getFunctionKey](#) and [pin\\_getEncryptedData:minLength:maxLength:messageID:language:](#)

14.9.2.71 - (RETURN\_CODE) pin\_getAmount: (int) *minLength* maxLength:(int) *maxLength* messageID:(int) *mID*  
language:(LANGUAGE\_TYPE) *lang*

##### Display Message and Get Amount

Decrypt and display message on LCD. Requires secure message data. Returns value in inputValue of [deviceDelegate::pinpadData:keySN:event:\(\)](#) with event EVENT\_PINPAD\_AMOUNT

#### Parameters

<i>minLength</i>	Minimum account number length - not less than 1
<i>maxLength</i>	Maximum account number length - not more than 16
<i>mID</i>	Message ID from approved message list.
<i>lang</i>	Language file to use for message  <pre>typedef enum{ LANGUAGE_TYPE_ENGLISH, LANGUAGE_TYPE_PORTUGUESE, LANGUAGE_TYPE_SPANISH }LANGUAGE_TYPE;</pre>

#### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString:</a>
--------------------	--

#### Notes

- If there is no any enter in 3 minutes, this command will time out.
- If there is no any enter in 20 seconds, the entered namount key will be cleared.
- When press Enter key , it will end this Command and response package with NGA format .
- When press Cancel key, the entered amount key will be cleared and if press Cancel key again, this command terminated.



- Cancel Command can terminate this command.

### Secure Messages

Secure messages to be used with General Prompts commands

Msg Id	English Prompt	Portuguese Prompt	Spanish Prompt
1	ENTER	ENTER	INGRESE
2	REENTER	RE-INTRODUZIR	REINGRESE
3	ENTER YOUR	INTRODUZIR O SEU	INGRESE SU
4	REENTER YOUR	RE-INTRODUZIR O SEU	REINGRESE SU
5	PLEASE ENTER	POR FAVOR DIGITE	POR FAVOR INGRESE
6	PLEASE REENTER	POR FAVO REENTRAR	POR FAVO REINGRESE
7	PO NUMBER	NÚMERO PO	NUMERO PO
8	DRIVER ID	LICENÇA	LICENCIA
9	ODOMETER	ODOMETER	ODOMETRO
10	ID NUMBER	NÚMERO ID	NUMERO ID
11	EQUIP CODE	EQUIP CODE	CODIGO EQUIP
12	DRIVERS ID	DRIVER ID	ID CONDUCTOR
13	JOB NUMBER	EMP NÚMERO	NUMERO EMP
14	WORK ORDER	TRABALHO ORDEM	ORDEN TRABAJO
15	VEHICLE ID	ID VEÍCULO	ID VEHICULO
16	ENTER DRIVER	ENTER DRIVER	INGRESE CONDUCTOR
17	ENTER DEPT	ENTER DEPT	INGRESE DEPT
18	ENTER PHONE	ADICIONAR PHONE	INGRESE TELEFONO
19	ENTER ROUTE	ROUTE ADD	INGRESE RUTA
20	ENTER FLEET	ENTER FROTA	INGRESE FLOTA
21	ENTER JOB ID	ENTER JOB ID	INGRESE ID TRABAJO
22	ROUTE NUMBER	NÚMERO PATH	RUTA NUMERO
23	ENTER USER ID	ENTER USER ID	INGRESE ID USUARIO
24	FLEET NUMBER	NÚMERO DE FROTA	FLOTA NUMERO
25	ENTER PRODUCT	ADICIONAR PRODUTO	INGRESE PRODUCTO
26	DRIVER NUMBER	NÚMERO DRIVER	CONDUCTOR NUMERO
27	ENTER LICENSE	ENTER LICENÇA	INGRESE LICENCIA
28	ENTER FLEET NO	ENTER NRO FROTA	INGRESE NRO FLOTA
29	ENTER CAR WASH	WASH ENTER	INGRESE LAVADO
30	ENTER VEHICLE	ENTER VEÍCULO	INGRESE VEHICULO
31	ENTER TRAILER	TRAILER ENTER	INGRESE TRAILER
32	ENTER ODOMETER	ENTER ODOMETER	INGRESE ODOMETRO
33	DRIVER LICENSE	CARTEIRA DE MOTORISTA	LICENCIA CONDUCTOR
34	ENTER CUSTOMER	ENTER CLIENTE	INGRESE CLIENTE
35	VEHICLE NUMBER	NÚMERO DO VEÍCULO	VEHICULO NUMERO
36	ENTER CUST DATA	ENTER CLIENTE INFO	INGRESE INFO CLIENTE

37	REENTER DRIVID	REENTRAR DRIVER ID	REINGRESE ID CHOFER
38	ENTER USER DATA	ENTER INFO USUÁRIO	INGRESE INFO USUARIO
39	ENTER CUST CODE	ENTER CODE. CLIENTE	INGRESE COD. CLIENTE
40	ENTER EMPLOYEE	ENTER FUNCIONÁRIO	INGRESE EMPLEADO
41	ENTER ID NUMBER	ENTER NÚMERO ID	INGRESE NUMERO ID
42	ENTER DRIVER ID	ENTER ID DRIVER	INGRESE ID CONDUCTOR
43	ENTER FLEET PIN	ENTER PIN FROTA	INGRESE PIN DE FLOTA
44	ODOMETER NUMBER	NÚMERO ODOMETER	ODOMETRO NUMERO
45	ENTER DRIVER LIC	ENTER DRIVER LIC	INGRESE LIC CONDUCTOR
46	ENTER TRAILER NO	NRO TRAILER ENTER	INGRESE NRO TRAILER
47	REENTER VEHICLE	REENTRAR VEÍCULO	REINGRESE VEHICULO
48	ENTER VEHICLE ID	ENTER VEÍCULO ID	INGRESE ID VEHICULO
49	ENTER BIRTH DATE	INSERIR DATA NAC	INGRESE FECHA NAC
50	ENTER DOB MMDDYY	ENTER FDN MMDDYY	INGRESE FDN MMDDAA
51	ENTER FLEET DATA	ENTER FROTA INFO	INGRESE INFO DE FLOTA
52	ENTER REFERENCE	ENTER REFERÊNCIA	INGRESE REFERENCIA
53	ENTER AUTH NUMBR	ENTER NÚMERO AUT	INGRESE NUMERO AUT
54	ENTER HUB NUMBER	ENTER HUB NRO	INGRESE NRO HUB
55	ENTER HUBOMETER	MEDIDA PARA ENTRAR HUB	INGRESE MEDIDO DE HUB
56	ENTER TRAILER ID	TRAILER ENTER ID	INGRESE ID TRAILER
57	ODOMETER READING	QUILOMETRAGEM	LECTURA ODOMETRO
58	REENTER ODOMETER	REENTRAR ODOMETER	REINGRESE ODOMETRO
59	REENTER DRIV. ID	REENTRAR DRIVER ID	REINGRESE ID CHOFER
60	ENTER CUSTOMER ID	ENTER CLIENTE ID	INGRESE ID CLIENTE
61	ENTER CUST. ID	ENTER CLIENTE ID	INGRESE ID CLIENTE
62	ENTER ROUTE NUM	ENTER NUM ROUTE	INGRESE NUM RUTA
63	ENTER FLEET NUM	FROTA ENTER NUM	INGRESE NUM FLOTA
64	FLEET PIN	FROTA PIN	PIN DE FLOTA
65	DRIVER #	DRIVER #	CONDUCTOR #
66	ENTER DRIVER #	ENTER DRIVER #	INGRESE CONDUCTOR #
67	VEHICLE #	VEÍCULO #	VEHICULO #
68	ENTER VEHICLE #	ENTER VEÍCULO #	INGRESE VEHICULO #
69	JOB #	TRABALHO #	TRABAJO #
70	ENTER JOB #	ENTER JOB #	INGRESE TRABAJO #
71	DEPT NUMBER	NÚMERO DEPT	NUMERO DEPTO

72	DEPARTMENT #	DEPARTAMENTO #	DEPARTAMENTO #
73	ENTER DEPT #	ENTER DEPT #	INGRESE DEPTO #
74	LICENSE NUMBER	NÚMERO DE LICENÇA	NUMERO LICENCIA
75	LICENSE #	LICENÇA #	LICENCIA #
76	ENTER LICENSE #	ENTER LICENÇA #	INGRESE LICENCIA #
77	DATA	INFO	INFO
78	ENTER DATA	ENTER INFO	INGRESE INFO
79	CUSTOMER DATA	CLIENTE INFO	INFO CLIENTE
80	ID #	ID #	ID #
81	ENTER ID #	ENTER ID #	INGRESE ID #
82	USER ID	USER ID	ID USUARIO
83	ROUTE #	ROUTE #	UTA #
84	ENTER ROUTE #	ADD ROUTE #	INGRESE RUTA #
85	ENTER CARD NUM	ENTER NÚMERO DE CARTÃO	INGRESE NUM TARJETA
86	EXP DATE(Yymm)	VALIDADE VAL (AAMM)	FECHA EXP (AAMM)
87	PHONE NUMBER	TELEFONE	NUMERO TELEFONO
88	CVV START DATE	CVV DATA DE INÍCIO	CVV FECHA INICIO
89	ISSUE NUMBER	NÚMERO DE EMISSÃO	NUMERO DE EMISION
90	START DATE (MMYY)	DATA DE INÍCIO (AAMM)	FECHA INICIO (AAMM)

14.9.2.72 - (RETURN\_CODE) pin\_getCardAccount: (int) *minLength* max:(int) *maxLength* line1:(NSString \*) *line1* line2:(NSString \*) *line2*

Display Message and Get Card Account

Show message on LCD and get card account number from keypad, then return encrypted card account number. Returns encryptedData of entered account in deviceDelegate::pinpadData:keySN:event:() with event EVENT\_PINPAD\_ACCOUNT

Parameters

<i>minLength</i>	Minimum account number length - not less than 1
<i>maxLength</i>	Maximum account number length - not more than 16
<i>line1</i>	Display line 1, up to 12 characters
<i>line2</i>	Display line 2, up to 16 characters

Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

Notes

- If there is no any enter in 3 minutes, this command time out.
- If there is no any enter in 20 seconds, the entered account numbers will be cleared.
- When press Enter key, it will end this command and respond package with NGA format.
- When press Cancel key, the entered account numbers will be cleared and if press Cancel key again, this command terminated.
- Cancel command can terminate this command.

14.9.2.73 - (RETURN\_CODE) pin\_getEncryptedData: (BOOL) *lastPackage* minLength:(int) *minLength* maxLength:(int) *maxLength* messageId:(int) *mID* language:(LANGUAGE\_TYPE) *lang*

Display Message and Get Encrypted Data

Decrypt and display message on LCD. Prompts the user with up to 2 lines of text. Returns value of encrypted data (using MSR DUKPT key) and KSN to deviceDelegate::pinpadData:keySN:event:() with event EVENT\_PINPAD\_ENCRYPTED\_DATA

#### Parameters

<i>lastPackage</i>	Last package flag
<i>minLength</i>	Minimum account number length - not less than 1
<i>maxLength</i>	Maximum account number length - not more than 16
<i>mID</i>	Message ID from approved message list.
<i>lang</i>	Language file to use for message  <pre>typedef enum{ LANGUAGE_TYPE_ENGLISH, LANGUAGE_TYPE_PORTUGUESE, LANGUAGE_TYPE_SPANISH }LANGUAGE_TYPE;</pre>

#### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### Notes

- If there is no any enter in 3 minutes, this command time out.
- If there is no any enter in 20 seconds, the entered account numbers will be cleared.
- When press Enter key, it will end this command and respond package with NGA format.
- When press Cancel key, the entered account numbers will be cleared and if press Cancel key again, this command terminated.
- Cancel command can terminate this command.
- Maximum pattern number allowed is 10. If any error or invalid command is sent, input data patterns will be cleared and command fail.

#### Secure Messages

Secure messages to be used with General Prompts commands

Msg Id	English Prompt	Portuguese Prompt	Spanish Prompt
1	ENTER	ENTER	INGRESE
2	REENTER	RE-INTRODUZIR	REINGRESE
3	ENTER YOUR	INTRODUZIR O SEU	INGRESE SU
4	REENTER YOUR	RE-INTRODUZIR O SEU	REINGRESE SU
5	PLEASE ENTER	POR FAVOR DIGITE	POR FAVOR INGRESE
6	PLEASE REENTER	POR FAVO REENTRAR	POR FAVO REINGRESE
7	PO NUMBER	NÚMERO PO	NUMERO PO
8	DRIVER ID	LICENÇA	LICENCIA
9	ODOMETER	ODOMETER	ODOMETRO

10	ID NUMBER	NÚMERO ID	NUMERO ID
11	EQUIP CODE	EQUIP CODE	CODIGO EQUIP
12	DRIVERS ID	DRIVER ID	ID CONDUCTOR
13	JOB NUMBER	EMP NÚMERO	NUMERO EMP
14	WORK ORDER	TRABALHO ORDEM	ORDEN TRABAJO
15	VEHICLE ID	ID VEÍCULO	ID VEHICULO
16	ENTER DRIVER	ENTER DRIVER	INGRESE CONDUCTOR
17	ENTER DEPT	ENTER DEPT	INGRESE DEPT
18	ENTER PHONE	ADICIONAR PHONE	INGRESE TELEFONO
19	ENTER ROUTE	ROUTE ADD	INGRESE RUTA
20	ENTER FLEET	ENTER FROTA	INGRESE FLOTA
21	ENTER JOB ID	ENTER JOB ID	INGRESE ID TRABAJO
22	ROUTE NUMBER	NÚMERO PATH	RUTA NUMERO
23	ENTER USER ID	ENTER USER ID	INGRESE ID USUARIO
24	FLEET NUMBER	NÚMERO DE FROTA	FLOTA NUMERO
25	ENTER PRODUCT	ADICIONAR PRODUTO	INGRESE PRODUCTO
26	DRIVER NUMBER	NÚMERO DRIVER	CONDUCTOR NUMERO
27	ENTER LICENSE	ENTER LICENÇA	INGRESE LICENCIA
28	ENTER FLEET NO	ENTER NRO FROTA	INGRESE NRO FLOTA
29	ENTER CAR WASH	WASH ENTER	INGRESE LAVADO
30	ENTER VEHICLE	ENTER VEÍCULO	INGRESE VEHICULO
31	ENTER TRAILER	TRAILER ENTER	INGRESE TRAILER
32	ENTER ODOMETER	ENTER ODOMETER	INGRESE ODOMETRO
33	DRIVER LICENSE	CARTEIRA DE MOTORISTA	LICENCIA CONDUCTOR
34	ENTER CUSTOMER	ENTER CLIENTE	INGRESE CLIENTE
35	VEHICLE NUMBER	NÚMERO DO VEÍCULO	VEHICULO NUMERO
36	ENTER CUST DATA	ENTER CLIENTE INFO	INGRESE INFO CLIENTE
37	REENTER DRIVID	REENTRAR DRIVER ID	REINGRESE ID CHOFER
38	ENTER USER DATA	ENTER INFO USUÁRIO	INGRESE INFO USUARIO
39	ENTER CUST CODE	ENTER CODE. CLIENTE	INGRESE COD. CLIENTE
40	ENTER EMPLOYEE	ENTER FUNCIONÁRIO	INGRESE EMPLEADO
41	ENTER ID NUMBER	ENTER NÚMERO ID	INGRESE NUMERO ID
42	ENTER DRIVER ID	ENTER ID DRIVER	INGRESE ID CONDUCTOR
43	ENTER FLEET PIN	ENTER PIN FROTA	INGRESE PIN DE FLOTA
44	ODOMETER NUMBER	NÚMERO ODOMETER	ODOMETRO NUMERO
45	ENTER DRIVER LIC	ENTER DRIVER LIC	INGRESE LIC CONDUCTOR
46	ENTER TRAILER NO	NRO TRAILER ENTER	INGRESE NRO TRAILER
47	REENTER VEHICLE	REENTRAR VEÍCULO	REINGRESE VEHICULO

48	ENTER VEHICLE ID	ENTER VEÍCULO ID	INGRESE ID VEHICULO
49	ENTER BIRTH DATE	INSERIR DATA NAC	INGRESE FECHA NAC
50	ENTER DOB MMDDYY	ENTER FDN MMDDYY	INGRESE FDN MMDDAA
51	ENTER FLEET DATA	ENTER FROTA INFO	INGRESE INFO DE FLOTA
52	ENTER REFERENCE	ENTER REFERÊNCIA	INGRESE REFERENCIA
53	ENTER AUTH NUMBR	ENTER NÚMERO AUT	INGRESE NUMERO AUT
54	ENTER HUB NUMBER	ENTER HUB NRO	INGRESE NRO HUB
55	ENTER HUBOMETER	MEDIDA PARA ENTRAR HUB	INGRESE MEDIDO DE HUB
56	ENTER TRAILER ID	TRAILER ENTER ID	INGRESE ID TRAILER
57	ODOMETER READING	QUILOMETRAGEM	LECTURA ODOMETRO
58	REENTER ODOMETER	REENTRAR ODOMETER	REINGRESE ODOMETRO
59	REENTER DRIV. ID	REENTRAR DRIVER ID	REINGRESE ID CHOFER
60	ENTER CUSTOMER ID	ENTER CLIENTE ID	INGRESE ID CLIENTE
61	ENTER CUST. ID	ENTER CLIENTE ID	INGRESE ID CLIENTE
62	ENTER ROUTE NUM	ENTER NUM ROUTE	INGRESE NUM RUTA
63	ENTER FLEET NUM	FROTA ENTER NUM	INGRESE NUM FLOTA
64	FLEET PIN	FROTA PIN	PIN DE FLOTA
65	DRIVER #	DRIVER #	CONDUCTOR #
66	ENTER DRIVER #	ENTER DRIVER #	INGRESE CONDUCTOR #
67	VEHICLE #	VEÍCULO #	VEHICULO #
68	ENTER VEHICLE #	ENTER VEÍCULO #	INGRESE VEHICULO #
69	JOB #	TRABALHO #	TRABAJO #
70	ENTER JOB #	ENTER JOB #	INGRESE TRABAJO #
71	DEPT NUMBER	NÚMERO DEPT	NUMERO DEPTO
72	DEPARTMENT #	DEPARTAMENTO #	DEPARTAMENTO #
73	ENTER DEPT #	ENTER DEPT #	INGRESE DEPTO #
74	LICENSE NUMBER	NÚMERO DE LICENÇA	NUMERO LICENCIA
75	LICENSE #	LICENÇA #	LICENCIA #
76	ENTER LICENSE #	ENTER LICENÇA #	INGRESE LICENCIA #
77	DATA	INFO	INFO
78	ENTER DATA	ENTER INFO	INGRESE INFO
79	CUSTOMER DATA	CLIENTE INFO	INFO CLIENTE
80	ID #	ID #	ID #
81	ENTER ID #	ENTER ID #	INGRESE ID #
82	USER ID	USER ID	ID USUARIO
83	ROUTE #	ROUTE #	RUTA #
84	ENTER ROUTE #	ADD ROUTE #	INGRESE RUTA #
85	ENTER CARD NUM	ENTER NÚMERO DE CARTÃO	INGRESE NUM TARJETA
86	EXP DATE(Yymm)	VALIDADE VAL (Aamm)	FECHA EXP (Aamm)
87	PHONE NUMBER	TELEFONE	NUMERO TELEFONO

88	CVV START DATE	CVV DATA DE INÍCIO	CVV FECHA INICIO
89	ISSUE NUMBER	NÚMERO DE EMISSÃO	NUMERO DE EMISION
90	START DATE (MMYY)	DATA DE INÍCIO (AAMM)	FECHA INICIO (AAMM)

14.9.2.74 - (RETURN\_CODE) pin\_getEncryptedPIN: (NSString \*) account keyType:(PIN\_KEY\_Types) type line1:(NSString \*) line1 line2:(NSString \*) line2 line3:(NSString \*) line3

Display Message and Get Encrypted PIN online

Prompts the user with up to 3 lines of text. Returns pinblock/ksn of entered PIN value in deviceDelegate::pinpadData:keySN:event:() with event MessageID\_PINEntry

#### Parameters

<i>account</i>	Card account number
<i>type</i>	Encryption Key Type: <ul style="list-style-type: none"> <li>• 0x00: External Account Key PIN_KEY_TDES_MKSK_extp</li> <li>• 0x01: External Account Key PIN_KEY_TDES_DUKPT_extp</li> <li>• 0x10: Internal Account Key PIN_KEY_TDES_MKSK_intl</li> <li>• 0x11: Internal Account Key PIN_KEY_TDES_DUKPT_intl</li> </ul>
<i>line1</i>	Display line 1, up to 12 characters
<i>line2</i>	Display line 2, up to 16 characters
<i>line3</i>	Display line 3, up to 16 characters <ul style="list-style-type: none"> <li>•</li> </ul>

#### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

#### Notes

- If there is no any enter in 3 minutes, this command will time out.
- If there is no any enter in 20 seconds, the entered PIN key will be cleared.
- When press Enter key , it will end this Command and response package with NGA format .
- When press Cancel key, the entered PIN key will be cleared and if press Cancel key again, this command terminated.
- Cancel Command can terminate this command.

14.9.2.75 - (RETURN\_CODE) pin\_getFunctionKey

#### Get Function Key

Returns function key value of pressed key in deviceDelegate::pinpadData:keySN:event:() with event EVENT\_PINPAD\_FUNCTION\_KEY. Value passed as NSData in pinpadData with one of the following values

- 0x43: Cancel Key
- 0x42: Backspace Key
- 0x45: Enter Key

- 0x23: # Key
- 0x2A: \* Key



## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## Notes

- If there is no any enter in 3 minutes, this command will time out.
- Cancel Command can terminate this command.

14.9.2.76 - (RETURN\_CODE) pin\_getNumeric: (bool) *maskInput* minLength:(int) *minLength* maxLength:(int) *maxLength* messageId:(int) *mID* language:(LANGUAGE\_TYPE) *lang*

Display Message and Get Numeric Key(s)

Decrypt and display message on LCD. Requires secure message data. Returns value in inputValue of device← Delegate::pinpadData:keySN:event:() with event EVENT\_PINPAD\_NUMERIC

## Parameters

<i>maskInput</i>	If true, all entered data will be masked with asterik (*)
<i>minLength</i>	Minimum account number length - not less than 1
<i>maxLength</i>	Maximum account number length - not more than 16
<i>mID</i>	Message ID from approved message list.
<i>lang</i>	Language file to use for message  <pre>typedef enum{ LANGUAGE_TYPE_ENGLISH, LANGUAGE_TYPE_PORTUGUESE, LANGUAGE_TYPE_SPANISH }LANGUAGE_TYPE;</pre>

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## Notes

- If there is no any enter in 3 minutes, this command will time out.
- If there is no any enter in 20 seconds, the entered numeric key will be cleared.
- When press Enter key , it will end this Command and response package with NGA format .
- When press Cancel key, the entered numeric key will be cleared and if press Cancel key again, this command terminated.
- Cancel Command can terminate this command.

## Secure Messages

Secure messages to be used with General Prompts commands

Msg Id	English Prompt	Portuguese Prompt	Spanish Prompt
1	ENTER	ENTER	INGRESE
2	REENTER	RE-INTRODUZIR	REINGRESE
3	ENTER YOUR	INTRODUZIR O SEU	INGRESE SU

4	REENTER YOUR	RE-INTRODUZIR O SEU	REINGRESE SU
5	PLEASE ENTER	POR FAVOR DIGITE	POR FAVOR INGRESE
6	PLEASE REENTER	POR FAVO REENTRAR	POR FAVO REINGRESE
7	PO NUMBER	NÚMERO PO	NUMERO PO
8	DRIVER ID	LICENÇA	LICENCIA
9	ODOMETER	ODOMETER	ODOMETRO
10	ID NUMBER	NÚMERO ID	NUMERO ID
11	EQUIP CODE	EQUIP CODE	CODIGO EQUIP
12	DRIVERS ID	DRIVER ID	ID CONDUCTOR
13	JOB NUMBER	EMP NÚMERO	NUMERO EMP
14	WORK ORDER	TRABALHO ORDEM	ORDEN TRABAJO
15	VEHICLE ID	ID VEÍCULO	ID VEHICULO
16	ENTER DRIVER	ENTER DRIVER	INGRESE CONDUCTOR
17	ENTER DEPT	ENTER DEPT	INGRESE DEPT
18	ENTER PHONE	ADICIONAR PHONE	INGRESE TELEFONO
19	ENTER ROUTE	ROUTE ADD	INGRESE RUTA
20	ENTER FLEET	ENTER FROTA	INGRESE FLOTA
21	ENTER JOB ID	ENTER JOB ID	INGRESE ID TRABAJO
22	ROUTE NUMBER	NÚMERO PATH	RUTA NUMERO
23	ENTER USER ID	ENTER USER ID	INGRESE ID USUARIO
24	FLEET NUMBER	NÚMERO DE FROTA	FLOTA NUMERO
25	ENTER PRODUCT	ADICIONAR PRODUTO	INGRESE PRODUCTO
26	DRIVER NUMBER	NÚMERO DRIVER	CONDUCTOR NUMERO
27	ENTER LICENSE	ENTER LICENÇA	INGRESE LICENCIA
28	ENTER FLEET NO	ENTER NRO FROTA	INGRESE NRO FLOTA
29	ENTER CAR WASH	WASH ENTER	INGRESE LAVADO
30	ENTER VEHICLE	ENTER VEÍCULO	INGRESE VEHICULO
31	ENTER TRAILER	TRAILER ENTER	INGRESE TRAILER
32	ENTER ODOMETER	ENTER ODOMETER	INGRESE ODOMETRO
33	DRIVER LICENSE	CARTEIRA DE MOTORISTA	LICENCIA CONDUCTOR
34	ENTER CUSTOMER	ENTER CLIENTE	INGRESE CLIENTE
35	VEHICLE NUMBER	NÚMERO DO VEÍCULO	VEHICULO NUMERO
36	ENTER CUST DATA	ENTER CLIENTE INFO	INGRESE INFO CLIENTE
37	REENTER DRIVID	REENTRAR DRIVER ID	REINGRESE ID CHOFER
38	ENTER USER DATA	ENTER INFO USUÁRIO	INGRESE INFO USUARIO
39	ENTER CUST CODE	ENTER CODE. CLIENTE	INGRESE COD. CLIENTE
40	ENTER EMPLOYEE	ENTER FUNCIONÁRIO	INGRESE EMPLEADO
41	ENTER ID NUMBER	ENTER NÚMERO ID	INGRESE NUMERO ID
42	ENTER DRIVER ID	ENTER ID DRIVER	INGRESE ID CONDUCTOR

43	ENTER FLEET PIN	ENTER PIN FROTA	INGRESE PIN DE FLOTA
44	ODOMETER NUMBER	NÚMERO ODOMETER	ODOMETRO NUMERO
45	ENTER DRIVER LIC	ENTER DRIVER LIC	INGRESE LIC CONDUCTOR
46	ENTER TRAILER NO	NRO TRAILER ENTER	INGRESE NRO TRAILER
47	REENTER VEHICLE	REENTRAR VEÍCULO	REINGRESE VEHICULO
48	ENTER VEHICLE ID	ENTER VEÍCULO ID	INGRESE ID VEHICULO
49	ENTER BIRTH DATE	INSERIR DATA NAC	INGRESE FECHA NAC
50	ENTER DOB MMDDYY	ENTER FDN MMDDYY	INGRESE FDN MMDDAA
51	ENTER FLEET DATA	ENTER FROTA INFO	INGRESE INFO DE FLOTA
52	ENTER REFERENCE	ENTER REFERÊNCIA	INGRESE REFERENCIA
53	ENTER AUTH NUMBR	ENTER NÚMERO AUT	INGRESE NUMERO AUT
54	ENTER HUB NUMBER	ENTER HUB NRO	INGRESE NRO HUB
55	ENTER HUBOMETER	MEDIDA PARA ENTRAR HUB	INGRESE MEDIDO DE HUB
56	ENTER TRAILER ID	TRAILER ENTER ID	INGRESE ID TRAILER
57	ODOMETER READING	QUILOMETRAGEM	LECTURA ODOMETRO
58	REENTER ODOMETER	REENTRAR ODOMETER	REINGRESE ODOMETRO
59	REENTER DRIV. ID	REENTRAR DRIVER ID	REINGRESE ID CHOFER
60	ENTER CUSTOMER ID	ENTER CLIENTE ID	INGRESE ID CLIENTE
61	ENTER CUST. ID	ENTER CLIENTE ID	INGRESE ID CLIENTE
62	ENTER ROUTE NUM	ENTER NUM ROUTE	INGRESE NUM RUTA
63	ENTER FLEET NUM	FROTA ENTER NUM	INGRESE NUM FLOTA
64	FLEET PIN	FROTA PIN	PIN DE FLOTA
65	DRIVER #	DRIVER #	CONDUCTOR #
66	ENTER DRIVER #	ENTER DRIVER #	INGRESE CONDUCTOR #
67	VEHICLE #	VEÍCULO #	VEHICULO #
68	ENTER VEHICLE #	ENTER VEÍCULO #	INGRESE VEHICULO #
69	JOB #	TRABALHO #	TRABAJO #
70	ENTER JOB #	ENTER JOB #	INGRESE TRABAJO #
71	DEPT NUMBER	NÚMERO DEPT	NUMERO DEPTO
72	DEPARTMENT #	DEPARTAMENTO #	DEPARTAMENTO #
73	ENTER DEPT #	ENTER DEPT #	INGRESE DEPTO #
74	LICENSE NUMBER	NÚMERO DE LICENÇA	NUMERO LICENCIA
75	LICENSE #	LICENÇA #	LICENCIA #
76	ENTER LICENSE #	ENTER LICENÇA #	INGRESE LICENCIA #
77	DATA	INFO	INFO
78	ENTER DATA	ENTER INFO	INGRESE INFO
79	CUSTOMER DATA	CLIENTE INFO	INFO CLIENTE
80	ID #	ID #	ID #

81	ENTER ID #	ENTER ID #	INGRESE ID #
82	USER ID	USER ID	ID USUARIO
83	ROUTE #	ROUTE #	RUTA #
84	ENTER ROUTE #	ADD ROUTE #	INGRESE RUTA #
85	ENTER CARD NUM	ENTER NÚMERO DE CARTÃO	INGRESE NUM TARJETA
86	EXP DATE(YYYMM)	VALIDADE VAL (AAMM)	FECHA EXP (AAMM)
87	PHONE NUMBER	TELEFONE	NUMERO TELEFONO
88	CVV START DATE	CVV DATA DE INÍCIO	CVV FECHA INICIO
89	ISSUE NUMBER	NÚMERO DE EMISSÃO	NUMERO DE EMISION
90	START DATE (MMYY)	DATA DE INÍCIO (AAMM)	FECHA INICIO (AAMM)

#### 14.9.2.77 - (RETURN\_CODE) pin\_getPinLength: (NSData \*\*) response

Get PIN Length

Returns encrypted PIN Length

##### Parameters

<i>response</i>	Returned Value of Minimum/Maximum PIN length: <ul style="list-style-type: none"> <li>• response[0] = minimum length</li> <li>• response[1] = maximum length</li> </ul>
-----------------	--

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

```

NSData* res;
RETURN_CODE rt = [[IDT_BTPay sharedController] pin_getPinLength:&res];
uint8_t b[res.length];
[data getBytes:b];
if (RETURN_CODE_DO_SUCCESS == rt && res.length>1){
    LOGI(@"GetPinLength: min=%d max=%d", b[0], b[1]);
}

```

#### 14.9.2.78 - (RETURN\_CODE) pin\_getPINpadStatus: (NSData \*\*) response

Get PINpad Status

Returns status of PINpad

##### Parameters

<i>response</i>	Pinpad status. response[0]: <ul style="list-style-type: none"> <li>• 0x01: PINpad is inactivate.</li> <li>• 0x02: PINpad has been activated, but Public Key is not loaded.</li> <li>• 0x03: Public key has been loaded, but Firmware Key, Numeric Key and Check Value is not loaded.</li> <li>• 0x10: PINpad normal work status.</li> <li>• 0x30: PINpad suspend status if password input error.</li> <li>• 0x31: PINpad suspend status if get PIN(MKSK)120 times in one hours.</li> </ul>
-----------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

```

Byte *b = (Byte*)malloc(1);
RETURN_CODE rt = [[IDT_BTPay sharedController] pin_getPINpadStatus];
if (RETURN_CODE_DO_SUCCESS == rt){
    [self appendMessageToResults:[NSString stringWithFormat:@"PINPad Status: %d", b[0]]];
}

}

```

14.9.2.79 - (RETURN\_CODE) pin\_setPinLength: (int) *minLength* maxLength:(int) *maxLength*

## Set PIN Length

Sets the encrypted PIN length.

## Parameters

<i>minLength</i>	Minimum PIN length at least 4
<i>maxLength</i>	Maximum PIN length not to exceed 12

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.9.2.80 + (NSString\*) SDK\_version

## SDK Version

Returns the current version of IDTech.framework

## Return values

<i>Framework</i>	version
------------------	---------

## 14.9.2.81 + (IDT\_BTPay\*) sharedController

## Singleton Instance

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

## Return values

<i>Instance</i>	of <a href="#">IDT_BTPay</a>
-----------------	------------------------------

## 14.9.3 Property Documentation

## 14.9.3.1 - (id&lt; IDT\_BTPay\_Delegate &gt;) delegate [read],[write],[atomic],[strong]

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

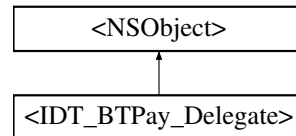
The documentation for this class was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/IDT\_BTPay.h

## 14.10 <IDT\_BTPay\_Delegate> Protocol Reference

```
#import <IDT_BTPay.h>
```

Inheritance diagram for <IDT\_BTPay\_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) - [dataInOutMonitor:incoming:](#)
- (void) - [swipeMSRData:](#)
- (void) - [emvTransactionData:errorCode:](#)
- (void) - [pinpadData:keySN:event:](#)

### 14.10.1 Detailed Description

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

### 14.10.2 Method Documentation

14.10.2.1 - (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>

14.10.2.2 - (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

**14.10.2.3** - (void) pinpadData: (NSData \*) value keySN:(NSData \*) KSN event:(EVENT\_PINPAD\_Types) event [optional]

Pinpad data delegate protocol

Receives data from pinpad methods

## Parameters

<i>value</i>	encrypted data returned from <a href="#">pin_getEncryptedData:minLength:maxLength:messageID:language: (IDT_BTPay)</a> , or encrypted account number returned from <a href="#">pin_getCardAccount:max:line1:line2: (IDT_BTPay)</a> . String value returned from <a href="#">pin_getAmount:maxLength:messageID:language: (IDT_BTPay)</a> or <a href="#">pin_getNumeric:minLength:maxLength:messageID:language: (IDT_BTPay)</a> . PINblock returned from <a href="#">pin_getEncryptedPIN:keyType:line1:line2:line3: (IDT_BTPay)</a>
<i>KSN</i>	Key Serial Number returned from <a href="#">pin_getEncryptedPIN:keyType:line1:line2:line3: (IDT_BTPay)</a> , <a href="#">pin_getCardAccount:max:line1:line2: (IDT_BTPay)</a> or <a href="#">pin_getEncryptedData:minLength:maxLength:messageID:language: (IDT_BTPay)</a>
<i>event</i>	EVENT_PINPAD_Types PINpad event that solicited the data capture

```
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;
```

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

Receives card data from MSR swipe.

## Parameters

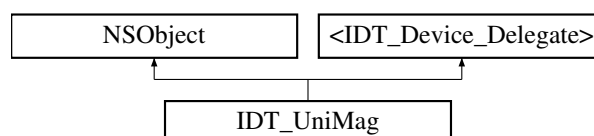
<i>cardData</i>	Captured card data from MSR swipe
-----------------	-----------------------------------

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

- /Users/randy/Repo/IDT/IDTech/IDT\_BTPay.h

## 14.11 IDT\_UniMag Class Reference

Inheritance diagram for IDT\_UniMag:



## Instance Methods

- (RETURN\_CODE) - [device\\_sendUniMagCommand:](#)
- (RETURN\_CODE) - [config\\_getSerialNumber:](#)
- (NSString \*) - [device\\_getResponseCodeString:](#)
- (RETURN\_CODE) - [device\\_connectToAudioReader](#)
- (RETURN\_CODE) - [msr\\_cancelMSRSwipe](#)
- (RETURN\_CODE) - [msr\\_startMSRSwipe](#)
- (RETURN\_CODE) - [msr\\_setSwipeEncryption:](#)
- (RETURN\_CODE) - [device\\_setPrePANClearDigits:](#)
- (RETURN\_CODE) - [msr\\_getNextKSN:](#)
- (RETURN\_CODE) - [msr\\_setSwipeForcedEncryptionOption:](#)
- (bool) - [isConnected](#)

## Class Methods

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

## Properties

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

### 14.11.1 Method Documentation

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

Polls device for Serial Number

##### Parameters

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

##### Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString:</a>
--------------------	--

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

Connect To Audio Reader

##### Return values

<i>RETURN_CODE</i>	Attempts to recognize and connect to an IDTech MSR device connected via the audio port.
--------------------	---

#### 14.11.1.3 - (NSString \*) device\_getResponseCodeString: (int) errorCode

Get Response Code String

Interpret a [IDT\\_UniMag](#) response code and return string description.



## Parameters

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

## Return values

<i>Verbose</i>	error description
----------------	-------------------

14.11.1.4 - (RETURN\_CODE) device\_sendUniMagCommand: (UNIMAG\_COMMAND\_Types) *command*

## Send UniMag Command

## Parameters

<i>command</i>	A command to execute from UNIMAG_COMMAND_Types enumeration
----------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.11.1.5 - (RETURN\_CODE) device\_setPrePANClearDigits: (int) *clearDigits*

## Set PrePAN Clear Digits

Sets the number of digits to show in clear text at the beginning of PAN

## Parameters

<i>clearDigits</i>	Amount of characters to display cleartext at beginning of PAN. Valid range 0-6. Default value 4.
--------------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> ;
--------------------	---

## 14.11.1.6 - (bool) isConnected

## Device Connected

## Return values

<i>isConnected</i>	Boolean indicated if UniMag is connected
--------------------	--

## 14.11.1.7 - (RETURN\_CODE) msr\_cancelMSRSwipe

## Disable MSR Swipe

Cancels MSR swipe request.

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.11.1.8 - (RETURN\_CODE) msr\_getNextKSN: (NSData \*\*) *response*

Polls device for next KSN

## Parameters

<i>response</i>	Returns next KSN
-----------------	------------------

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.11.1.9 - (RETURN\_CODE) msr\_setSwipeEncryption: (int) *encryption*

## Set Swipe Data Encryption

Sets the swipe encryption method

## Parameters

<i>encryption</i>	1 = TDES, 2 = AES
-------------------	-------------------

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

14.11.1.10 - (RETURN\_CODE) msr\_setSwipeForcedEncryptionOption: (BOOL) *forceON*

## Set Swipe Force Encryption

Sets the swipe force encryption options

## Parameters

<i>forceON</i>	TRUE = Force Encryption ON, FALSE = Force Encryption OFF
----------------	--

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.11.1.11 - (RETURN\_CODE) msr\_startMSRSwipe

## Enable MSR Swipe

Enables MSR, waiting for swipe to occur. Allows track selection. Returns [IDTMSRData](#) instance to `deviceDelegate->::swipeMSRData:()`

## Return values

<i>RETURN_CODE</i>	Return codes listed as typedef enum in <a href="#">IDTCommon:RETURN_CODE</a> . Values can be parsed with <a href="#">device_getResponseCodeString</a> :
--------------------	---

## 14.11.1.12 + (NSString\*) SDK\_version

## SDK Version

- All Devices

Returns the current version of IDTech.framework

## Return values

<i>Framework</i>	version
------------------	---------

## 14.11.1.13 + (IDT\_UniMag\*) sharedController

## Singleton Instance

- All Devices

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

## Return values

<i>Instance</i>	of <a href="#">IDT_UniMag</a>
-----------------	-------------------------------

## 14.11.2 Property Documentation

## 14.11.2.1 -(id&lt; IDT\_UniMag\_Delegate &gt;) delegate [read],[write],[atomic],[strong]

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

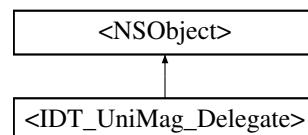
The documentation for this class was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/IDT\_UniMag.h

## 14.12 &lt;IDT\_UniMag\_Delegate&gt; Protocol Reference

```
#import <IDT_UniMag.h>
```

Inheritance diagram for <IDT\_UniMag\_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:](#)

## 14.12.1 Detailed Description

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

### 14.12.2 Method Documentation

14.12.2.1 - (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>

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

Receives messages from the framework

#### Parameters

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

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

Monitors the headphone jack for device insertion/removal.

#### Parameters

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

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

Receives card data from MSR swipe.

#### Parameters

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

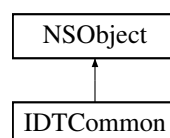
The documentation for this protocol was generated from the following file:

- /Users/andy/Repo/IDT/IDTech/IDT\_UniMag.h

## 14.13 IDTCommon Class Reference

```
#import <IDTCommon.h>
```

Inheritance diagram for IDTCommon:



### 14.13.1 Detailed Description

Common structures and enumerations shared by IDTech Framework classes. Evaluate header source for more info

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

- /Users/randy/Repo/IDT/IDTech/IDTCommon.h

## 14.14 IDTech\_ApplicationID Struct Reference

```
#include <IDTechEMV.h>
```

### Data Fields

- 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 [applicationVersionNumber](#) [2]  
*EMV application version number. Tag 9F09.*
- unsigned char [tacDefault](#) [5]  
*Terminal Action Code Denial. Tag DF13.*
- unsigned char [tacDenial](#) [5]  
*Terminal Action Code Denial. Tag DF14.*
- unsigned char [tacOnline](#) [5]  
*Terminal Action Code Denial. Tag DF15.*
- unsigned char [transactionCurrencyCode](#) [2]  
*AID's currency. Example: For US, {0x08,0x40}. Tag 5F2A.*
- unsigned char [transactionCurrencyExponent](#)  
*Transaction Currency Exponent. Example: Amount \$4.53 is managed as 453. Tag 5F36.*
- unsigned char [useTACDefault](#)  
*Indicates if tacDefault value should be used.*
- unsigned char [useTACDenial](#)  
*Indicates if tacDefault value should be used.*
- unsigned char [useTACOnline](#)  
*Indicates if tacDefault value should be used.*
- unsigned char [applicationSelectionIndicator](#)  
*Indicates if partial AID matching is allowed. 0x01 = allowed DF62.*

### 14.14.1 Detailed Description

device AID File - 47 bytes

Used as parameter in [setApplicationData: \(IDTechEMV\)](#)

Used as return value of aidResponse in [retrieveApplicationData: \(IDTechEMV\)](#)

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

- /Users/randy/Repo/IDT/IDTech/IDTechEMV.h

## 14.15 IDTech\_TerminalData Struct Reference

```
#include <IDTechEMV.h>
```

### Data Fields

- unsigned char [terminalCountryCode](#) [2]  
*Terminal's location. Tag 9F1A {0x08,0x40}.*
- unsigned char [merchantCategoryCode](#) [2]  
*Classifies the type of business being done by the merchant, represented according to ISO 8583:1993.*
- unsigned char [merchantID](#) [15]  
*When concatenated with the Acquirer Identifier, uniquely identifies a given merchant.*
- unsigned char [terminalID](#) [8]  
*Designates the unique location of a terminal at a merchant.*
- unsigned char [defaultTACDefault](#) [5]  
*Default Terminal Action Code Denial. Tag DF13.*
- unsigned char [defaultTACDenial](#) [5]  
*Default Terminal Action Code Denial. Tag DF14.*
- unsigned char [defaultTACOnline](#) [5]  
*Default Terminal Action Code Denial. Tag DF15.*
- unsigned char [useDefaultTACDefault](#)  
*Indicates if tacDefault value should be used as terminal default value.*
- unsigned char [useDefaultTACDenial](#)  
*Indicates if tacDenial value should be used as terminal default value.*
- unsigned char [useDefaultTACOnline](#)  
*Indicates if tacOnline value should be used as terminal default value.*
- unsigned char [terminalLocation](#) [36]  
*Terminal Location tag 9F4E.*

### 14.15.1 Detailed Description

device Terminal Configuration File - 45 bytes

Used as parameter in [setTerminalData: \(IDTechEMV\)](#)

Used as return value in [setTerminalData: \(IDTechEMV\)](#)

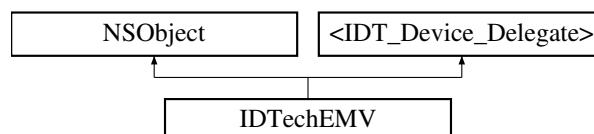
The documentation for this struct was generated from the following file:

- /Users/andy/Repo/IDT/IDTech/IDTechEMV.h

## 14.16 IDTechEMV Class Reference

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

Inheritance diagram for IDTechEMV:



## Instance Methods

- (void) - [completeOnlineEMVTransaction:resultCode:issuerAuthenticationData:issuerScripts:](#)
- (void) - [confirmApplication:](#)
- (void) - [confirmApplicationCancel](#)
- (NSDictionary \*) - [getAllTags](#)
- (NSString \*) - [getEMVKernelVersion](#)
- (NSData \*) - [getTag:](#)
- (void) - [removeApplicationData:](#)
- (NSArray \*) - [retrieveAIDList](#)
- (IDTech\_ApplicationID) - [retrieveApplicationData:](#)
- (IDTech\_TerminalData) - [retrieveTerminalData](#)
- (void) - [setApplicationData:](#)
- (void) - [setTag:tagData:](#)
- (void) - [preloadEMVCardData](#)
- (void) - [setTerminalData:](#)
- (void) - [cancelTransaction](#)
- (void) - [startEMVTransaction:timeout:transactionType:additionalTags:](#)
- (void) - [accelerateRead:](#)
- (void) - [sendSystemError](#)

## Class Methods

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

## Properties

- id< [IDTechEMV\\_Delegate](#) > [delegate](#)

### 14.16.1 Detailed Description

Class to drive the [IDTechEMV](#) device

### 14.16.2 Method Documentation

#### 14.16.2.1 - (void) [accelerateRead:](#) (BOOL) *enabled*

Accelerate Read Data

Enables multi-record reading from ICC to accelerate EMV transaction time. Enabled by default.

Parameters

<i>enabled</i>	TRUE = use accelerated reading, FALSE = use standard reading
----------------	--

#### 14.16.2.2 - (void) [cancelTransaction](#)

Cancel Transaction

Cancels the current transaction



**14.16.2.3** - (void) completeOnlineEMVTransaction: (EMV\_COMPLETION\_RESULT) *result* resultCode:(NSString \*) *rc* issuerAuthenticationData:(NSString \*) *iad* issuerScripts:(NSString \*) *scripts*

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>result</i>	EMV_COMPLETION_RESULT: Used to specify if contacting online host was successful or other problem occurred
<i>rc</i>	Result Code from host. Mandatory. 2 characters ASCII value. Example "00"
<i>iad</i>	Issuer Authentication Data. Optional. 10 bytes, 20 Hex Characters representing data. Example "11223344556677883030"
<i>scripts</i>	Issuer Scripts. Optional. Data represented by Hex Characters. TLV Format. Must start with 71 or 72, followed by length, followed by data. Example "711000112233445566778899AABBCCDDEEFF". Can string multiple scripts, both 71 and 72.

Results are returned on delegate protocol emvTransactionData.

**14.16.2.4** - (void) confirmApplication: (int) *index*

Set Application

During and EMV transaction flow, if there are multiple applications to chose from, or the terminal settings indicate cardholder confirmation for applicaiton selection, the delegate IDTechEMV\_Delegate::confirmApplicationSelection:() will receive an array with all the available applications to choose from. The selected index of the application must be passed back to this method to continue the EMV transaction flow

Parameters

<i>index</i>	The index of the selected app from the application array passed back from confirmApplicationSelection:()
--------------	--

**14.16.2.5** - (void) confirmApplicationCancel

Cancel Set Application

During and EMV transaction flow, if there are multiple applications to chose from, or the terminal settings indicate cardholder confirmation for applicaiton selection, the delegate IDTechEMV\_Delegate::confirmApplicationSelection:() will receive an array with all the available applications to choose from. If no application selection is performed, this routine must be called to cancel the transaction

**14.16.2.6** - (NSDictionary\*) getAllTags

Get All Tags

Retrieves all EMV tags from the inserted card. Only available after the card has been processed after executing IDTechEMV::startEMVTransaction:timeout:additionalTags:()

Return values

<i>data</i>	NSDictionary with all tag values. Key is NSString, Object is NSData.
-------------	--

**14.16.2.7** - (NSString\*) getEMVKernelVersion

Returns SDK EMV Kernel Version

## Return values

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

## 14.16.2.8 - (NSData\*) getTag: (NSString \*) tagName

## Get Tag

Retrieves an EMV tag from the inserted card. Only available after the card has been processed after executing IDTechEMV::startEMVTransaction:timeout:additionalTags:()

## Parameters

<i>tagName</i>	Name fo tag to retrieve
----------------	-------------------------

## Return values

<i>data</i>	Tag data
-------------	----------

## 14.16.2.9 - (void) preloadEMVCardData

## Preload EMV Card Data

This function will attempt to cache preliminary EMV card data. It is to be used in an environment where there will be a few seconds between inserting the card and executing the startEMV command. Example would be the idle time in collecting an amount before startEVM is executed.

## 14.16.2.10 - (void) 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
------------	---

## 14.16.2.11 - (NSArray\*) retrieveAIDList

## Retrieve AID list

Returns all the AID name/length on the inserted ICC. Populates response parameter with an dictionary with Keys of AID Names (NSData\*), and values of AID Lengths (NSData\*)

## Return values

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

## 14.16.2.12 - (IDTech\_ApplicationID) retrieveApplicationData: (NSString \*) AID

## Retrieve Application Data by AID

Retrieves the Application Data as specified by the AID name passed as a parameter. The AID will be in the response parameter responseAID. If aidLen = 0, then requested AID was not found.

## Parameters

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

## Return values

<i>responseAID</i>	The AID returned from the method
--------------------	----------------------------------

## 14.16.2.13 - (IDTech\_TerminalData) retrieveTerminalData

## Retrieve Terminal Data

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

## Return values

<i>response</i>	Response returned as a <a href="#">TerminalData</a>
-----------------	---

## 14.16.2.14 - (void) sendSystemError

## Send Sytem Error

For Internal Use Only

## 14.16.2.15 - (void) setApplicationData: (IDTech\_ApplicationID) data

## Set Application Data by AID

Sets the Application Data as specified by the [ApplicationID](#) structure passed as a parameter

## Parameters

<i>data</i>	<a href="#">ApplicationID</a> configuration file
-------------	--

## 14.16.2.16 - (void) setTag: (NSString \*) tagName tagData:(NSData \*) data

## Set Tag

Sets an EMV tag from the tags read into the terminal. Only available after the card has been processed after executing IDTechEMV::startEMVTransaction:timeout:additionalTags:()

## Parameters

<i>tagName</i>	Name of tag to set
<i>data</i>	Tag data

## 14.16.2.17 - (void) setTerminalData: (IDTech\_TerminalData) data

## Set Terminal Data

Sets the Terminal Data as specified by the [IDTech\\_TerminalData](#) structure passed as a parameter

## Parameters

<i>data</i>	<a href="#">TerminalData</a> configuration file
-------------	---

#### 14.16.2.18 + (IDTechEMV\*) sharedController

Singleton Instance

Establishes an singleton instance of [IDTechEMV](#) class.

Return values

<i>Instance</i>	of <a href="#">IDTechEMV</a>
-----------------	------------------------------

#### 14.16.2.19 - (void) startEMVTransaction: (float) amount timeout:(int) timeout transactionType:(int) type additionalTags:(NSMutableDictionary \*) tags

Start EMV Transaction Request

Authorizes the EMV transaction amounts for an ICC card

The tags will be returned in the emvTransactionData delegate protocol.

Parameters

<i>amount</i>	Transaction amount value (tag value 9F02)
<i>timeout</i>	Timeout value in seconds.
<i>type</i>	Transaction Type.
<i>tags</i>	Any other optional tags to be included in the request. Passed as a mutable dictionary.

#### Converting TLV to NSMutableDictionary

EMV data is received in TLV (Tag, Length, value) format: 950500000080009B02E8009F2701018A025A339F26080C552B9364D55CE5

This data contains the following EMV tags/values:

Tag	Length	Value
95	05	0000008000
9B	02	E800
9F27	01	01
8A	02	5A33
9F26	08	0C552B9364D55CE5

An example how to create an NSMutableDictionary with these values follows.

```
-(NSMutableDictionary*) createTLVDict{
NSMutableDictionary *emvTags = [[NSMutableDictionary alloc] initWithCapacity:0];

[emvTags setObject:@"0000008000" forKey:@"95"];
[emvTags setObject:@"E800" forKey:@"9B"];
[emvTags setObject:@"01" forKey:@"9F27"];
[emvTags setObject:@"5A33" forKey:@"8A"];
[emvTags setObject:@"0C552B9364D55CE5" forKey:@"9F26"];

return emvTags;
}
```

### 14.16.3 Property Documentation

14.16.3.1 - (id< IDTechEMV\_Delegate >) delegate [read],[write],[atomic],[strong]

- Reference to [IDTechEMV\\_Delegate](#).

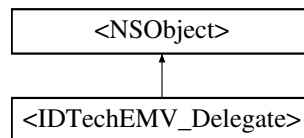
The documentation for this class was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/IDTechEMV.h

## 14.17 <IDTechEMV\_Delegate> Protocol Reference

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

Inheritance diagram for <IDTechEMV\_Delegate>:



### Instance Methods

- (void) - [confirmApplicationSelection:retry:](#)
- (void) - [languagePreference:](#)
- (void) - [emvTransactionData:errorCode:performReversal:](#)
- (void) - [emvTransactionMessage:](#)
- (void) - [swipeMSRDataEMV:emv:](#)

*Receives card data from MSR swipe.*

### 14.17.1 Detailed Description

Protocol methods established for [IDTechEMV](#) class

### 14.17.2 Method Documentation

14.17.2.1 - (void) confirmApplicationSelection: (NSArray \*) labelArray retry:(BOOL) tryAgain [optional]

Receives available applications to select from during and EMV transactions when terminal is set for cardholder confirmation or a decision must be made between multiple applications available for selection. Array index of selected app must be passed back to [confirmApplication: \(IDTechEMV\)](#) to continue the EMV transaction flow.

Parameters

<i>labelArray</i>	Array of available application labels. Use this array item index to pass back to <a href="#">confirmApplication: (IDTechEMV)</a>
<i>tryAgain</i>	Flag to indicate requirement to display "Try Again" along with application selection.

14.17.2.2 - (void) emvTransactionData: (IDTEMVData \*) emvData errorCode:(int) error performReversal:(BOOL) reversal [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
<i>reversal</i>	If this flag is TRUE, this means the transaction was APPROVED online, but a TC was not returned by the kernel and the transaction was declined. In most cases, the online host needs to be contacted to reverse the approval.

14.17.2.3 - (void) emvTransactionMessage: (MESSAGE\_Types) *message* [optional]

## EMV Messages

During an EMV Transaction flow , this delegate with pass messages that to request further interaction, or post information

## Parameters

<i>message</i>	EMV Message Type-  <pre>typedef enum{     MESSAGE_INSERT_CARD = 0,     MESSAGE_REMOVE_CARD } EMV_MESSAGE_Types;</pre>
----------------	---

14.17.2.4 - (void) languagePreference: (NSData \*) *lang* [optional]

## Language Preference

After the kernel loads the Application, it will send back the Language Preference, tag 5F2D. This can be used to determine what language to use when displaying prompts and messages for the remainder of the transaction flow.

## Parameters

<i>lang</i>	1-4 languages stored in order of preference, each represented by 2 alphabetical characters according to ISO 639.
-------------	--

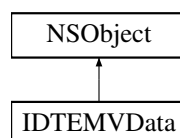
The documentation for this protocol was generated from the following file:

- /Users/andy/Repo/IDT/IDTech/IDTechEMV.h

## 14.18 IDTEMVData Class Reference

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

Inheritance diagram for IDTEMVData:



## Instance Methods

- (void) - [clear](#)

## Class Methods

- (IDTEMVData \*) + sharedController

## Properties

- EMV\_RESULT\_CODE\_Types resultCode
- int encryptionMode  
0 = TDES, 1 = AES
- int cardType  
0 = Contact, 1 = Contactless
- 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.

### 14.18.1 Detailed Description

Encapsulating data class for EMV data capture

### 14.18.2 Method Documentation

#### 14.18.2.1 - (void) clear

clears all IDTEMVData properties

#### 14.18.2.2 + (IDTEMVData \*) sharedController

Singleton instance of IDTEMVData

### 14.18.3 Property Documentation

#### 14.18.3.1 - (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;

```

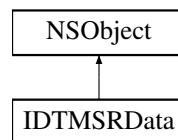
The documentation for this class was generated from the following file:

- /Users/randy/Repo/IDT/IDTech/IDTEMVData.h

## 14.19 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.*

### 14.19.1 Detailed Description

Encapsulating data class for MSR data capture

### 14.19.2 Method Documentation

#### 14.19.2.1 - (void) clear

clears all [IDTMSRData](#) properties

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

Singleton instance of [IDTMSRData](#)

### 14.19.3 Property Documentation

#### 14.19.3.1 - (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;
```

### 14.19.3.2 - (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;
```

### 14.19.3.3 - (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_Types;
```

### 14.19.3.4 - (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:

- /Users/andy/Repo/IDT/IDTech/IDTMSRData.h

## 14.20 MaskAndEncryption Struct Reference

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

### Data Fields

- 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](#)

### 14.20.1 Detailed Description

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

### 14.20.2 Field Documentation

#### 14.20.2.1 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

#### 14.20.2.2 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:

- /Users/randy/Repo/IDT/IDTech/IDTCommon.h

## 14.21 PowerOnStructure Struct Reference

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

### Data Fields

- BOOL [sendIFS](#)  
*Send S(IFS) request if T=1 protocolError: Reference source not found.*
- BOOL [explicitPPS](#)  
*Explicit PPSError: Reference source not found.*
- BOOL [disableAutoPPS](#)  
*No auto pps for negotiate mode.*
- BOOL [disableResponseCheck](#)  
*No check on response of S(IFS) request.*
- unsigned char \* [pps](#)  
*pps is used to set the Protocol and Parameters Selection between card and reader, only Di <= 4 are supported. pps must follow the structure specified in ISO 7816-3 as PPS0, [PPS1], [PPS2], and [PPS3]. For more information see ISO 7816-3 section 7.2.*
- unsigned char [ppsLength](#)  
*length of pps data*

### 14.21.1 Detailed Description

Structure to set ICC power on options. Used by `icc_powerOnICC:response: (IDT_BTPay) IDT_UniPay::icc_power↔OnICC:response:()`

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

- `/Users/randy/Repo/IDT/IDTech/IDTCommon.h`

## 14.22 TerminalData Struct Reference

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

### Data Fields

- 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.*

### 14.22.1 Detailed Description

device Terminal Configuration File - 44 bytes

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

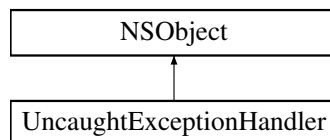
Used as return value in `emv_retrieveTerminalData: (IDT_BTPay)`

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

- `/Users/randy/Repo/IDT/IDTech/IDTCommon.h`

## 14.23 UncaughtExceptionHandler Class Reference

Inheritance diagram for UncaughtExceptionHandler:



### Protected Attributes

- `BOOL dismissed`

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

- `/Users/randy/Repo/IDT/IDTech/UncaughtExceptionHandler.h`