



ID TECH Payment Agent Application Command Reference

**Rev H
21 March 2022**

ID TECH
10721 Walker Street
Cypress, CA 90630-4720
Tel: (714) 761-6368 Fax: (714) 761-8880
www.idtechproducts.com

Copyright© 2022 International Technologies and Systems Corporation. All rights reserved.

ID TECH
10721 Walker Street
Cypress, CA 90630 USA

This document, as well as the software and hardware described in it, is furnished under license and may be used or copied online in accordance with the terms of such license. The content of this document is furnished for information use only, is subject to change without notice, and should not be construed as a commitment by ID TECH. While every effort has been made to ensure the accuracy of the information provided, ID TECH assumes no responsibility or liability for any unintentional errors or inaccuracies that may appear in this document. Except as permitted by such license, no part of this publication may be reproduced or transmitted by electronic, mechanical, recording, or otherwise, or translated into any language form without the express written consent of ID TECH.

ID TECH and ViVOpay are trademarks or registered trademarks of ID TECH.

Warranty Disclaimer

Software, services, and hardware are provided "as is" and "as-available" and the use of the software, services, and hardware are at the user's own risk. ID TECH does not make, and hereby disclaims, any and all other express or implied warranties, including, but not limited to, warranties of merchantability, fitness for a particular purpose, title, and any warranties arising from a course of dealing, usage, or trade practice. ID TECH does not warrant that the services or hardware will be uninterrupted, error-free, or completely secure.

CONFIDENTIAL

Table of Contents

1. OVERVIEW	5
1.1. Note	5
2. PA COMMAND OVERVIEW	6
2.1. Message Format	6
2.2. Message Field Details	6
2.3. Response Codes	6
2.4. Error Checking	7
3. PA COMMANDS	8
3.1. CFU: Perform Configuration Update	8
3.1.1. Configuration Fields	8
3.2. CPT: Cancel Payment Transaction	9
3.3. DPT: Do Payment Transaction	9
3.3.1. Payment Fields	9
3.4. GCS: Get Current State	10
3.4.1. GCS Response Fields	10
3.5. GOT: Get Offline Transactions	11
3.5.1. Payment Fields	12
3.6. GTI: Get Terminal Info	12
3.6.1. GTI Response Fields	12
3.7. RBT: Reboot Device	12
4. STORE & FORWARD	13
5. APPENDIX A: RESPONSE CODES	14
6. APPENDIX B: CRC-16 SAMPLE CODE	17
7. APPENDIX C: ERROR CONVERSION SAMPLE CODE	18
8. APPENDIX D: SIZE, DEFAULTS, MIN/MAX	19
9. APPENDIX E: STORE & FORWARD DIAGRAMS	20
9.1. Power On: Online	20
9.2. Power On: Offline/Store	21
9.3. Background: Forward	22

Revision History

Rev	Date	Description	Author
50	11/01/2018	Initial Release	CF+RB
51	11/05/2018	Added Overall Solution Diagram Fixed typos	CF
52	07/03/2019	Major Changes	CF
53	07/08/2019	Formatting and typographical corrections	RB
54	07/02/2020	Added SWC command, updated DPT response codes, updated available config values, other misc clean up	MH
55	07/04/2020	Changes to command structure, meaning/usage of certain fields, and the command terminator	MH
56	08/01/2020	Changed SWC to CPT, documented JSON changes, added response code table to index. Other general formatting.	MH
57	08/20/2020	Updating response codes; adding information on new code format.	MH
58	08/24/2020	Adding information on min/Maximum values and Maximum sizes	MH
59	08/26/2020	Fixing mistakes in Appendix A (response codes)	MH
60	08/31/2020	Added response codes; updated code in Appendix C;	MH
61	10/05/2020	Added response codes; Added GCS; Update config vales	MH
62	11/09/2020	Added response codes	MH
A	11/18/2020	Style update, first formal release.	CB
B	12/21/2020	Updated for v1.3: Store & Forward	MH
C	02/10/2021	Added additional response codes	MH
D	04/01/2021 06/09/2021	Added additional response codes Review and release, improved readability of Appendix A: Response Codes table	MH CB
E	7/20/2021	Added new commands; Modified response codes (added, removed, moved)	MH
F	10/26/2021	Added response code	MH
G	02/15/2022	Added new CFU commands and update default timeout	SB
H	03/21/2022	Expand on GTI and GCS response	SB

1. Overview

Payment Agent is a piece of software that resides in a VP5300. It sits between a POS system and the ITSCO server backend. The ITSCO server then facilitates communication with the specified payment gateway.

Communication to the POS is via a set of commands that are sent to the VP5300 from the POS system via RS-232 or USB.

The PA allows ID TECH to offer a way to assist ISVs in development of their own payment applications by abstracting the SDK to a higher level and providing a software-independent way to interfacing with the SDK.

1.1. Note

This was previously called the PAE (Payment Application Engine) Command Specification. The name has been updated such that the PAE application is now the Agent application. With Agent there will be different functions and features supported. The first of those is the PA (Payment Application). The references to PAE in the commands themselves will remain untouched. As such, this is a change in name and nothing else.

CONFIDENTIAL

2. PA Command Overview

The section below provides details about PA commands.

2.1. Message Format

PA command messages are ASCII text strings that always conform to this format:

```
<header>|<command>|<message type>|<response code>|<data>|<checksum>|<terminator>
```

There are always seven fields with a variable number of data parameters. The command message fields are delimited by pipe characters. The data is represented as a single JSON object with one or more name-value pairs. Back-to-back delimiters are allowed and must be expected as they represent an empty field.

Note: PA treats all JSON information as case-insensitive.

2.2. Message Field Details

Field	Notes
<header>	*PAE
<command>	See PAE command list. For example, DPT. Case-sensitive, must be all uppercase.
<message type>	CMD (to PA) RESP (from PA) ERR (from PA)
<response code>	Command result code for RESP messages. See Appendix A – Response Codes for specific code information. Codes can also be found in C/C++ header file: <code>ResponseCodes.h</code>
<data>	JSON format with enclosing JSON object and comma separated name-value pairs.
<checksum>	See below.
<terminator>	*!PAE!*

2.3. Response Codes

Response codes on messages sent to PA are ignored. Commands should use an empty field or one-digit number for those commands.

Response codes from PA will always be eight hexadecimal characters. These eight characters are two error codes combined. The first four characters (two bytes) contain information on the general error. This field always has some hex value in it that can include all zeroes, which would indicate success. The second four characters (two bytes) contain the detail response. This field can and frequently will be all zeroes.

There are a couple of ways the POS implementer could isolate these codes into two variables. Option one would be using string handling functions to divide the eight-character string into two 4-character strings. From there the two variables could be compared as strings or converted to integer values for comparison. Option two would be done as follows. To get the first error code, convert the string to an integer and then right-shift that value by 16-bits. To get the second error code, do a bit wise AND of the

original converted integer and 0x0000FFFF. See [Appendix C: Error Conversion Sample Code](#) for example code of this method.

2.4. Error Checking

The checksum is the decimal representation of a CRC-16 (CRC-16/CCITT-FALSE) calculation. All messages to and from PA must include a checksum. See [Appendix B: CRC-16 Sample Code](#) for example code calculating the CRC.

The checksum is calculated on the PA command message up to and including the final pipe character preceding the checksum field.

PA confirms the checksum of all commands it receives. If the checksum does not match, PA responds with a “Bad CRC” error (see [Appendix A: Response Codes](#)).

CONFIDENTIAL

3. PA Commands

Message Code	Description
CFU	Configuration Update
CPT	Cancel Payment Transaction
DPT	Do Payment Transaction
GCS	Get Current Status
GOT	Get Offline Transactions
GTI	Get Terminal Info
RBT	Reboots the device

3.1. CFU: Perform Configuration Update

The **CFU** command updates the PA configuration stored in the reader. Transmit the configuration data as a JSON string.

Expected message format:

PAE|CFU|CMD|1|<Config Data>|<checksum>|!PAE!*

Response message format:

PAE|CFU|RESP|<Result Code>||<checksum>|!PAE!*

3.1.1. Configuration Fields

Configuration Parameter	Default Value	JSON Type	Min-Maximum Value/Length
NFC_Disabled	1 (true)	Number	0 or 1 only; EMV_Disabled = 1, this is ignored
EMV_Disabled	1 (true)	Number	0 or 1 only
CardCaptureTimeout	30 (seconds)	Number	Min: 30 (seconds); Max: 180 (seconds)
ITSCO_Req_Timeout	10000 (milliseconds)	Number	No limits; suggest < 1 min. Value is expressed in milliseconds. Thus 10 sec = 10000
Payment_Req_Timeout	45000 (milliseconds)	Number	No limits; suggest > 30 seconds. Value is expressed in milliseconds. Thus 45 sec = 45000
SF_Disabled	1 (true)	Number	0 or 1 only
SF_MaxAmt	0	Number	0+ ; note: this field is in pennies (100 = \$1.00)
SF_MaxTime	0	Number	0+ ; note: this field is expressed in hours (24 = 1 day)
Checkin_Delay	300 (seconds)	Number	60 second minimum

3.2. CPT: Cancel Payment Transaction

The **CPT** command cancels an active **DPT** command. The reader accepts this command *following* a **DPT** request, but *before* a payment card is swiped or inserted.

Expected message format:

PAE|CPT|CMD|1||<checksum>|!PAE!*

Response message format:

PAE|CPT|RESP|<Result Code>||<checksum>|!PAE!*

3.3. DPT: Do Payment Transaction

This command is used to transmit payment functions.

Expected message format:

PAE|DPT|CMD|1|<Payment Data>|<checksum>|!PAE!*

Response message format:

PAE|DPT|RESP|1|<Result Code>|<Response Data>|<checksum>|!PAE!*

3.3.1. Payment Fields

Payment Parameter	Required	Value Information (no defaults)	JSON Type	Maximum Value/Length
OrderID	No	Can be used to set a customer-specific order ID.	String	Length: 256 bytes
PaymentAmount	Yes	Integer value indicating payment amount in cents (for example, 1.00 = 100).	Number	Min: 1 (\$0.00)
PaymentType	Yes	1: Sale/Purchase 2: Purchase w/Cash back 3: Refund w/Card 4: Refund wo/Card 5: Account Verification 6: Auth Only	Number	Min: 1 Max: 6
PaymentUser	Yes	String used to identify different terminal attendants/users. Can always be the same value if desired.	String	Length 256 bytes

Note: Does not support **PaymentType** 2: Purchase w/Cash back or 4: Refund wo/Card.

3.4. GCS: Get Current State

The **GCS** command gets the state of different device properties.

Expected message format:

PAE|GCS|CMD|1||<checksum>|!PAE!*

Response message format:

PAE|GCS|RESP|0|<State Info>|checksum>|!PAE!*

3.4.1. GCS Response Fields

GCS <State Info>	Value Information (no defaults)	JSON Type	Maximum Value/Length
Started	On PAE App start, value is 0 and is changed to 1 immediately after PAE enables processing of *PAE commands	Number	Min: 1
Registered	On PAE App start, value is 0 and is changed to 1 after successful Application registration response from service.	Number	Min: 1
Connected	On PAE App start, value is 0. Value is changed to 0 after any network request returns a networking error (such as timeout, DNS error, TLS error, etc). Value is changed to 1 after any network request returns a successful response from the service (including declines and gateway errors).	Number	Min: 1
Encryption	On PAE App start, value is 1. Value is 0 at PAE initialization if device encryption is disabled or no data (card) key has been loaded.	Number	Min: 1
CardInReader	When GCS command is processed, PAE checks for a card seated in the ICC reader and reports 1 if a card is seated or 0 if a card is not seated.	Number	Min: 1
InPayment	On PAE App start, value is 0. The value is set to 1 when a DPT command begins to process and is set back to 0 when the DPT command completes.	Number	Min: 1
OfflineRecovery	On PAE App start, value is 0. The value is set to 1 when when stored transactions begin to forward and is set to 0 when the transaction forwarding stops. Note: forwarding may stop before all stored transactions have forwarded, such as if the device goes offline again.	Number	Min: 1
~OfflineTxnsAvail	A number, with a minimum length of 1 and a maximum length of 4 digits for the current devices (VP5300 and VP6800). The number is a conservative	Number	Min:1

	estimate of the transactions that can be stored in the device, based on the remaining available storage in the device. Note: as the stored transaction data is variable in size, this number will likely not decrease in direct relation to the number of transactions stored. See the GOT command to determine the number of currently stored transactions.		
AppLoaderStatus	<p>The string indicates the status of the AppLoader. The current values are:</p> <ul style="list-style-type: none"> • Startup • Contacting Cloud • Reconnect Delay • Validating • Downloading • Install Pending (*) • Installing (**) • Idle <p>(*) indicates that a device reboot (to install updates) will occur within the next 5 minutes (**) is typically not seen, as installation currently occurs only when PAE is not running (and, thus, is unavailable to respond to GCS commands)</p>	String	Min:4

3.5. GOT: Get Offline Transactions

The **GOT** command can retrieve general information on stored offline transaction or get specific payment information.

Expected message format (general information):

PAE|GOT|CMD|1||<checksum>|!PAE!*

Expected message format (specific payments):

PAE|GOT|CMD|1|<Offline Request Params>|<checksum>|!PAE!*

Response message format:

PAE|GOT|RESP|0|<Offline Info>|checksum>|!PAE!*

3.5.1. Payment Fields

Offline Transaction Parameter	Required	Value Information (no defaults)	JSON Type	Maximum Value/Length
page	Only for specific payment info	Selects the page number of payments to return. Can return up to 400 payments per page.	Number	Min: 1
paymentType	Only for specific payment info	Specify "EMV" or "MSR" to indicate which payments to return. Expect return to include up to 8192 bytes.	String	Length: 3

3.6. GTI: Get Terminal Info

The **GTI** command queries the terminal and returns information about the device and application.

Expected message format:

PAE|GTI|CMD|1||<checksum>|!PAE!*

Response message format:

PAE|GTI|RESP|0|<Terminal Data>|checksum>|!PAE!*

3.6.1. GTI Response Fields

GCS <Terminal Data>	Value Information (no defaults)	JSON Type	Maximum Value/Length
PAE_Version	Version of PAE loaded on the device	String	N/A
AppLoader_Version	Version of AppLoader loaded on the device	String	N/A
K81_FW_Version	K81 FW version loaded on the device	String	N/A
RT1050_FW_Version	RT1050 FW version loaded on the device	String	N/A
HW_Version	Payment hardware version	String	N/A
SerialNum	Device Serial Number	String	N/A
CompanyID	CompanyID assigned to the device	String	N/A
ApplicationID	ApplicationID assigned to the device	String	N/A

3.7. RBT: Reboot Device

The **RBT** command reboots the device.

Expected message format:

PAE|RBT|CMD|1||<checksum>|!PAE!*

Response message format:

PAE|RBT|RESP|0|<Error Information>|checksum>|!PAE!*

Note: No command response sent if successful.

4. Store & Forward

PA supports “Store & Forward” (S&F) for times when the device is offline. For PA to use S&F, the feature must be enabled and the maximum amount of charge and offline time must be configured.

The scenarios for the “store” portion are as follows:

1. If the device has gone offline via periodic health checks but receives a timeout in response to **DPT** commands, PA returns an error and sets the device into offline mode. The payment can be resent and will be processed as an offline payment.
2. The device is in offline mode. When the device receives a payment, PA checks for the following items:
 - S&F allowed
 - Time with offline payments < time allowed
 - Requested amount < amount allowed
 - There is space available on the device to store the payment

If these criteria are met, the card info is collected and stored. In addition, a **DPT** command returns an offline approval message as well as offline receipt information.

The scenarios for the “forward” portion are as follows:

1. When the device comes back online, it sends the stored transactions to the ITSCO server in the order received.
2. The device continues to process transactions as offline until it sends all the stored transactions.
3. The server responds with a signal to keep or delete the transaction just processed. If storage is required, the device stores that transaction until the next time it broadcasts offline transactions.

See [Appendix E: Store & Forward Diagrams](#) for flowcharts detailing Store & Forward scenarios.

5. Appendix A: Response Codes

Decimal Value	Hex Value	Name	Description
<i>General 0..199</i>			
0	0	PAE_COMMAND_SUCCESS	Command completed successfully.
1	1	PAE_COMMAND_FAILURE	Command failed to complete.
2	2	PAE_COMMAND_THROTTLED	Command was throttled (denied).
<i>Errors 200..399</i>			
<i>PAE Message Errors 200..209</i>			
200	C8	ERR_PAE_STARTING	Command refused because PAE is starting.
201	C9	ERR_BAD_MESSAGE	Message received is not valid.
202	CA	ERR_MISSING_TOKENS	Message received was missing fields.
203	CB	ERR_BAD_CRC	Message received has incorrect CRC.
204	CC	ERR_INVALID_CMD	Command requested does not exist.
205	CD	ERR_NULL_MESSAGE	PAE received an empty message.
<i>File System Errors 210..219</i>			
210	D2	ERR_FILE_CREATE	Can't create file on device.
211	D3	ERR_FILE_OPEN	Can't open file on device.
212	D4	ERR_FILE_READ	Can't read file on device.
213	D5	ERR_FILE_WRITE	Can't write file to device.
214	D6	ERR_FILE_DELETE	Can't remove file from device.
<i>Config Errors 220..229</i>			
220	DC	ERR_READING_CONFIG	Config can't be read.
221	DD	ERR_PARSE_CONFIG	Config can't be parsed.
222	DE	ERR_SAVE_CONFIG	Config can't be saved.
<i>Payment System Errors 230..249</i>			
230	E6	ERR_PAYMENT_START_FAILED	Device can't start EMV or MSR process.
<i>HTTP Errors 250..299</i>			
250	FA	ERR_REG_DEVICE	Failed to register device with ITSCO server.
251	FB	ERR_REG_APP	Failed to register PAE with ITSCO server.
252	FC	ERR_ITSCO_INTERNAL	Internal error on ITSCO server.
253	FD	ERR_COMPLETE_PAYMENT	Payment can't be finalized on ITSCO server.
254	FE	ERR_HTTP_SEND	Failed sending HTTP message.
255	FF	ERR_HTTP_RESP_TIMEOUT	Timed out waiting for HTTP response.
256	100	ERR_HTTP_RESP_ERROR	HTTP response was 400+ (client or server error).
257	101	ERR_MISSING_RESP_VALUE	Required response value missing in HTTP response.
258	102	ERR_SSL_CERTIFICATE_ERROR	Problem with the SSL cert on the device.
259	103	ERR_NO_NETWORK	No network connection detected.
260	104	ERR_HTTP_CLIENT_FAIL	Note: description for firmware errors intentionally omitted. If these errors are encountered contact IDT.
261	105	ERR_HTTP_CLIENT_FAIL_SOCKET	

Decimal Value	Hex Value	Name	Description
262	106	ERR_HTTP_CLIENT_FAIL_PARSE	
263	107	ERR_HTTP_CLIENT_FAIL_DNS	
264	108	ERR_HTTP_CLIENT_FAIL_ERROR	
265	109	ERR_HTTP_CLIENT_FAIL_TIMEOUT	
266	10A	ERR_HTTP_CLIENT_FAIL_CONN	
267	10B	ERR_HTTP_CLIENT_FAIL_CLOSED	
268	10C	ERR_HTTP_CLIENT_FAIL_MALLOC	
269	10D	ERR_HTTP_CLIENT_FAIL_PARAM	
270	10E	ERR_HTTP_CLIENT_FAIL_MUTEX	
271	10F	ERR_HTTP_CLIENT_FAIL_SSL	
272	110	ERR_HTTP_CLIENT_FAIL_CA	
273	111	ERR_HTTP_CLIENT_FAIL_SEND	
274	112	ERR_HTTP_CLIENT_FAIL_FILE	
275	113	ERR_HTTP_CLIENT_UNKNOWN	
General Errors 300..399			
300	12C	ERR_BAD_JSON	Invalid JSON.
301	12D	ERR_MEM_ALLOC	Device can't allocate memory.
302	12E	ERR_FIRMWARE	Firmware call failed.
303	12F	ERR_GENERAL	Grouping of errors not easily classified.
305	130	ERR_PARSE	PAE received a message with bad JSON.
306	131	ERR_MUTEX_FAILURE	PAE unable to create mutex.
307	132	ERR_ENC_NOT_ENABLED	Device does not have encryption enabled.
308	133	ERR_ENC_KEY_MISSING	Device does not have a data encryption key.
CFU 400..419			
		No specific CFU responses.	
CPT 420..439			
420	1A4	CPT_CANT_CANCEL	Payment can't be canceled (likely already sent).
421	1A5	CPT_NO_ACTIVE_PAYMENT	No active payment to cancel.
422	1A6	CPT_CANCEL_FAILED	Call to cancel payment failed.
GCS 440..459			
		No specific GCS responses.	
GTI 460..479			
460	1CC	GTI_PARTIAL_DATA	Not all data fields for GTI could be returned.
GOT 480..499			
480	1E0	GOT_NO_TYPE_SPECIFIED	EMV or MSR must be specified with paged data request
DPT 500..599			
500	1F4	DPT_ACCEPT	DPT command success.
501	1F5	DPT_DECLINE	DPT command failure.
502	1F6	DPT_MISSING_REQ_PARAM	Command received missing required parameter.

Decimal Value	Hex Value	Name	Description
503	1F7	DPT_INVALID_SUBCMD	DPT sub-command requested that does not exist.
504	1F8	DPT_BUSY	Device can only do one payment at a time.
505	1F9	DPT_TIMEOUT	Customer didn't provide card within required time.
506	1FA	DPT_BAD_READ	Card could not be read.
507	1FB	DPT_CANCELED	Transaction canceled by POS.
508	1FC	DPT_TERMINATED	Transaction failed due to no ACK.
509	1FD	DPT_FALLBACK_TO_CONTACT	
510	1FE	DPT_ERRORCODE	
511	1FF	DPT_UNKNOWN_RESPONSE	An unknown MSR response received from device.
512	200	DPT_EMV_FAILED	General EMV failure, timeout, etc.
513	201	DPT_LCD_MESSAGE	Pass through for LCD message. Data field will contain a JSON object with the LCD Code and Message.
514	202	DPT_UNSUPPORTED_AID	Card inserted has no supported AIDs.
515	203	DPT_BAD_READ_TERMINATING	Card read was bad and transaction terminated (can try again – initiated from POS).
516	204	DPT_BAD_READ_RETRYING	Card read was bad and PAE will attempt again.
517	205	DPT_SF_OUT_OF_STORAGE	Device has no offline storage remaining.
518	206	DPT_SF_NOT_ENABLED	Store & Forward is not enabled.
519	207	DPT_SF_AMT_ABOVE_ALLOWED	The payment amount is greater than allowed while offline.
520	208	DPT_SF_OFFLINE_TIME_EXCEEDED	The device has been offline longer than the max time configured for Store & Forward.
521	209	DPT_INVALID_AMOUNT	Payment amount requested is invalid (check for decimals).
522	20A	DPT_OFFLINE_ACCEPT	Device is offline but transaction was approved for Store & Forward.
523	20B	DPT_NO_ENCRYPTION	DPT failed because no encryption key is loaded.
RBT 600..610			
		No specific RBT responses.	

6. Appendix B: CRC-16 Sample Code

C/C++

```
const int CRCTable[] =
{
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7, 0x8108, 0x9129,
    0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF, 0x1231, 0x0210, 0x3273, 0x2252,
    0x52B5, 0x4294, 0x72F7, 0x62D6, 0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C,
    0xF3FF, 0xE3DE, 0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
    0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D, 0x3653, 0x2672,
    0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4, 0xB75B, 0xA77A, 0x9719, 0x8738,
    0xF7DF, 0xE7FE, 0xD79D, 0xC7BC, 0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861,
    0x2802, 0x3823, 0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
    0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12, 0xDBFD, 0xCBDC,
    0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A, 0x6CA6, 0x7C87, 0x4CE4, 0x5CC5,
    0x2C22, 0x3C03, 0x0C60, 0x1C41, 0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B,
    0x8D68, 0x9D49, 0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
    0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78, 0x9188, 0x81A9,
    0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F, 0x1080, 0x00A1, 0x30C2, 0x20E3,
    0x5004, 0x4025, 0x7046, 0x6067, 0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C,
    0xE37F, 0xF35E, 0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D, 0x34E2, 0x24C3,
    0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405, 0xA7DB, 0xB7FA, 0x8799, 0x97B8,
    0xE75F, 0xF77E, 0xC71D, 0xD73C, 0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676,
    0x4615, 0x5634, 0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3, 0xCB7D, 0xDB5C,
    0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A, 0x4A75, 0x5A54, 0x6A37, 0x7A16,
    0x0AF1, 0x1AD0, 0x2AB3, 0x3A92, 0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B,
    0x9DE8, 0x8DC9, 0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
    0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8, 0x6E17, 0x7E36,
    0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};

int CalcCRC(const char* buffer, unsigned int len)
{
    int crc = 0xFFFF;
    for (int i = 0; i < len; i++)
    {
        crc = CRCTable[((crc >> 8) ^ (buffer[i] & 0xFF))] ^ ((crc << 8) & 0xFFFF);
    }

    return crc;
}
```

7. Appendix C: Error Conversion Sample Code

C/C++

```
void DecodeResponse(char* responseCode, unsigned short* detailCode,  
                   unsigned short* generalCode)  
{  
    unsigned long tmp = (unsigned long)strtol(responseCode, NULL, 16);  
    *generalCode = tmp >> 16;  
    *detailCode = (unsigned short)(0x0000FFFF & tmp);  
}
```

CONFIDENTIAL

8. Appendix D: Size, Defaults, Min/Max

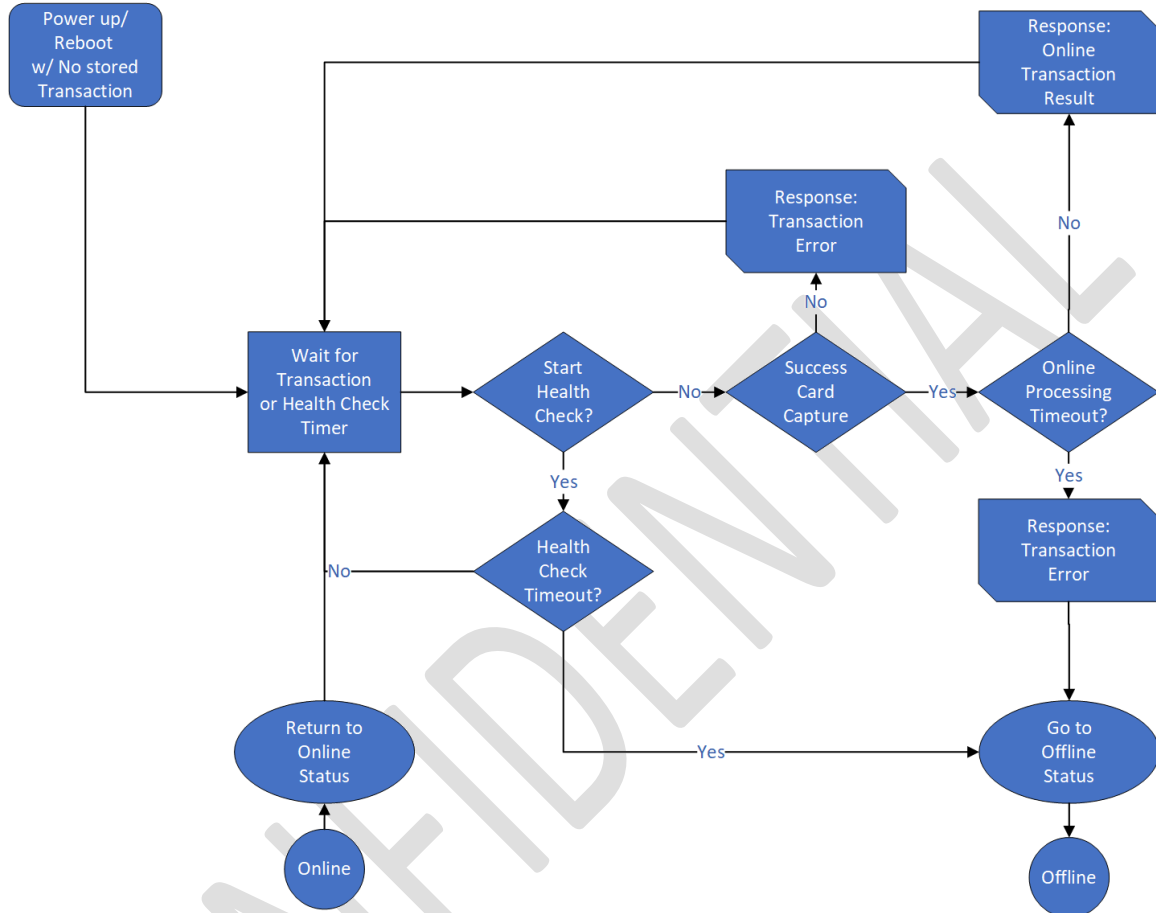
Name	Maximum Length	Notes
PAE message data field	8192 bytes	Maximum size of data field for command messages and responses.
DPT receipt data	1024 bytes	Maximum size of the receipt data within the DPT response.
JSON keys	256 bytes	Maximum length of any JSON key.
JSON string values	256 bytes	Maximum length of any JSON string value.
HTTP message body	16,384 bytes	Maximum amount of data that will be sent from device to ITSCO server (per call).
HTTP response body	16,384 bytes	Maximum amount of data that will be received by the device from the ITSCO server (per call).

CONFIDENTIAL

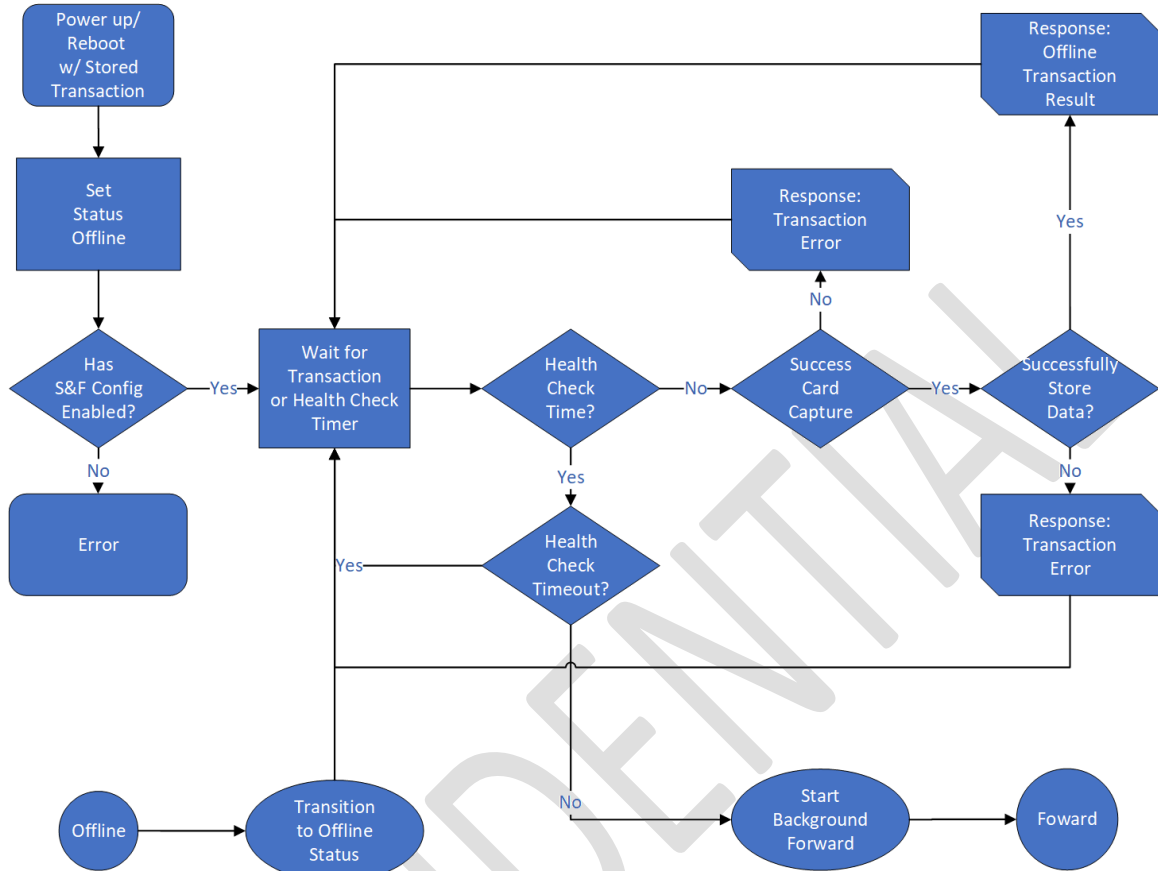
9. Appendix E: Store & Forward Diagrams

The following diagrams illustrate Store & Forward scenarios.

9.1. Power On: Online



9.2. Power On: Offline/Store



9.3. Background: Forward

