ADVANCED MOBILE PAYMENT INC.

AMP CONNECT

Developer Guide

v 4.0
INT-0100-4.0-E

PROPRIETARY NOTICE

All pages of this document contain information proprietary to Advanced Mobile Payment (AMP) Inc. This document shall not be duplicated, transmitted, used, or otherwise disclosed to anyone other than the organization or specific individuals to which this document is delivered. This restriction is applicable to all sheets of this document. AMP reserves the right to have the recipient return all copies of this document at any time. AMP reserves the right to change the content of this document without prior notice.

DOCUMENT PROPERTIES

INFORMATION

ID	INT-0100-4.0-E
Title	AMP Connect - Developer Guide
Document Portal Path	/Development/AMP POS/BaseApp Semi Integration/AMPConnect
Category	AMP POS
Access Level	Development
NDA Required	Yes ☑ No☐

VERSION CONTROL

Template Version	1.5

Doc. Version	Date	Summary of Change	Updated by
1.0	Aug 2019	Initial version	R. Vafaie
1.1	Nov 2019	Enhancements, including more supported transactions	B. Zhang N. Abouelsaad
1.2	Dec 2019	IOT app response data is added to transaction response data	B. Zhang
2.0	Mar 2020	Doc refactor, add new payment data format request support	B. Zhang
3.0	Apr 2020	Cloud based and tags refactoring	H. Saghari
4.0	Sep 2020	Added AMP Connect Agent	H. Saghari

SUPPORTED HARDWARE & SOFTWARE

Doc. Version	Software Title	Release	Supported Hardware Model
1.0	AMPConnect	V1.0	AMP POS 8 Series AMP POS 6700
1.1	AMPConnect	V2.5	AMP POS 8 Series AMP POS 6700

1.2	AMPConnect	V2.6.0	AMP POS 8 Series AMP POS 6700
1.2	AMPConnect Client	V2.2.0	AMP POS 8 Series
2.0	AMPConnect Client	V3.0	AMP POS 6, 8 Series
3.0	AMPConnect Client	V3.1	AMP POS 6, 8 Series
4.0	AMPConnect	V4.0.0+	AMP POS 6, 8 Series
4.0	AMPConnect Agent	V1.3.7+	AMP POS 6, 8 Series Any Android device

EXTERNAL REFERENCES

URLs	--
E-Mail Addresses	--
Phone Numbers	--
Documents	--
Knowledge Base Articles	--

TABLE OF CONTENTS

TABLE OF FIGURES

1 ABOUT AMP CONNECT

AMP Connect is a unified interface that enables a business application's seamless integration with AMP payment application, BaseApp.

The AMP Connect and AMP Connect Agent apps are middleware that enable communication between AMP payment terminal and another device that could be Windows, Linux, iOS or Android based. The communication method could be wired USB/Serial, wireless in the same network or wireless in the cloud. These applications enable access to BaseApp transactions or customer item display functionalities. Customer item display functionalities are utilized when AMP Connect is installed on an AMP 6700 terminal which is installed at the back of a Bridge device as a PINPad and customer display.

AMP Connect enables a third-party business application to perform all transactions that are supported by AMP payment application.

AMP Connect is hardware, host, and business agnostic. All messages are in JSON format which is a well-known industry standard. AMP Connect message format is unified for all communication methods and payment applications, attended or unattended.

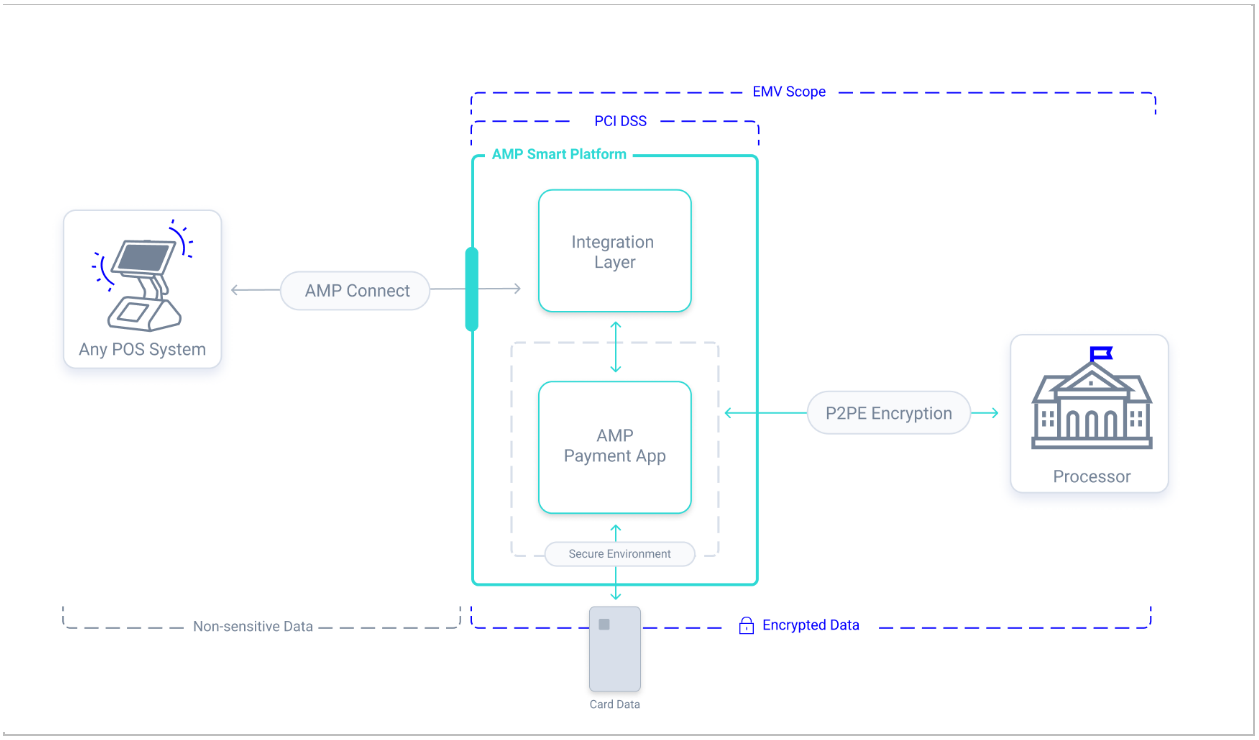


Figure 1 - AMP Connect Overview.

1.1 INTEGRATION CONSIDERATIONS

AMP Connect integration with a business app requires coordination with AMP. AMP assigns the integration support engineer to help with the project. Please contact your account manager to set up the integration project.

To be able to properly test the business app, AMP recommends using proper test cards. Test cards can be ordered through https://b2ps.com/products/b2-emv-test-card-sets/.

To ensure the proper test card set is selected, AMP recommendation is the following:

- For Canadian hosts:

UAT Canada EMV™ Test Card Set (16 cards)

- For US hosts:

UAT USA EMV™ Test Card Set (21 Cards)

USA Debit EMV™ Test Card Set (16 cards)

1.2 STARTING AND STOPPING THE AMP CONNECT APP

To start the AMP Connect app, launch the app from Android.

If the app is configured in auto launch mode, it will be launched automatically upon terminal restart.

To exit the app, click on the Android's back button. When asked for a password, enter the terminal serial number.

2 AMP CONNECT, AMP CONNECT AGENT, BUSINESS APPLICATION FORMATION

A business application utilizes AMP Connect and/or AMP Connect Agent to integrate to the AMP payment application or terminals and printer. Different formations are available to support different business application environments and requirements. Following are the different formations and benefits of each.

2.1 FORMATION 1

In this formation, business application runs on an external device. The external device platform is irrelevant. Business application communicates to AMP Connect through any media: WIFI, IP, USB, or Serial. The integration layer is media agnostic and can be used by any application on any platform.

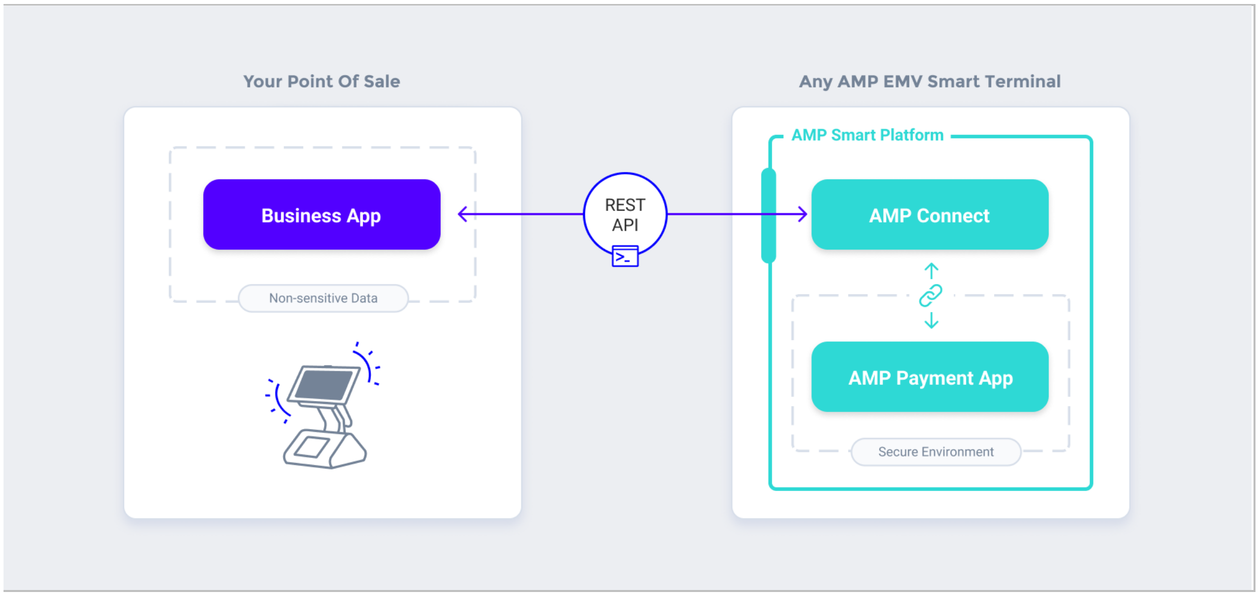


Figure 2 - Formation 1.

The business app can communicate with AMP Connect via AMP Cloud, as demonstrated in the following image.

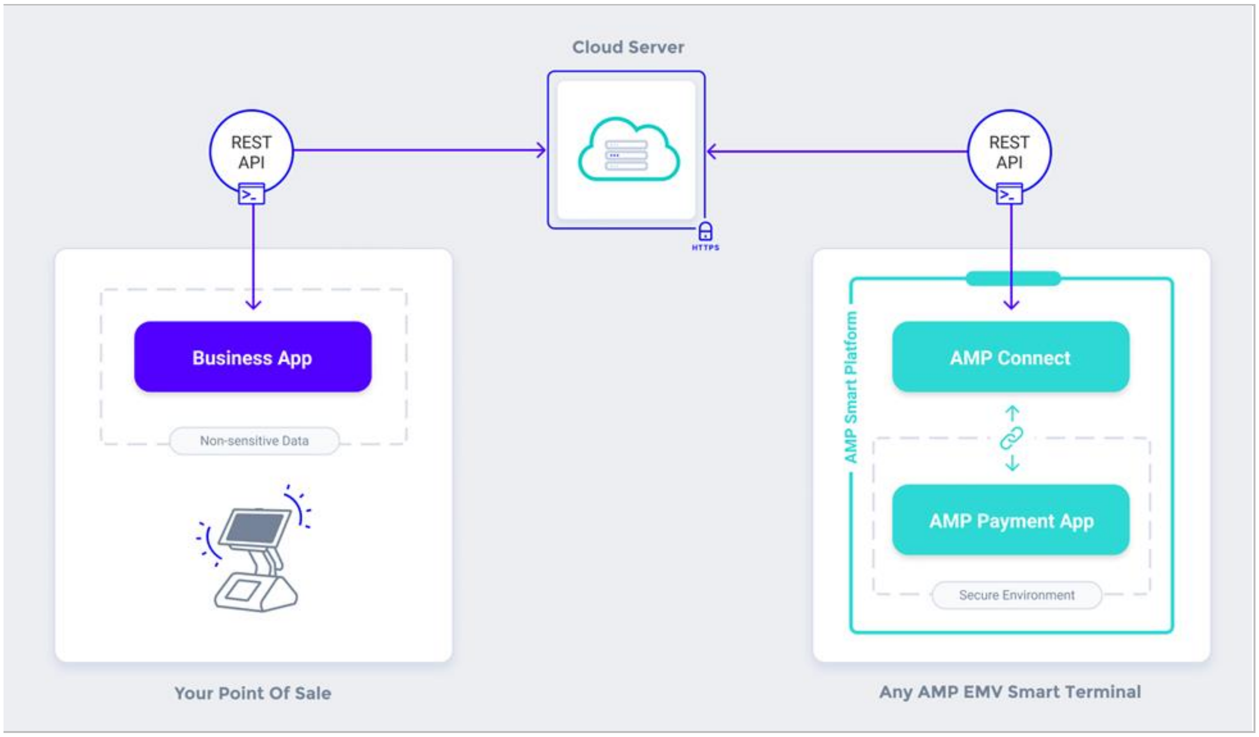


Figure 3 - Formation 1 through AMP Cloud.

2.2 FORMATION 2

In this formation, Business application runs on the Android terminal as a native application or a browser based application. Business app communicates to AMP payment app through AMP Connect Agent. The business app has access to the terminal printer in this formation.

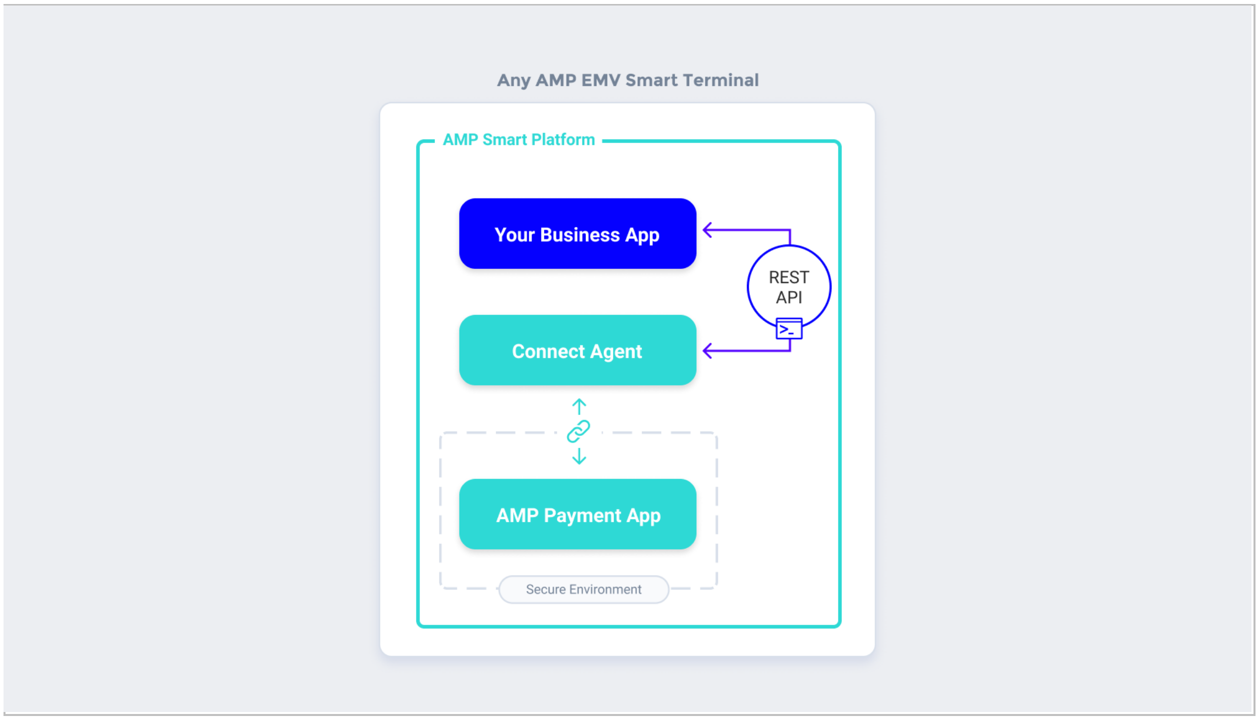


Figure 4 - Formation 2.

2.3 FORMATION 3

In this formation, business application is running on an android device and can be a browser based application. Business application can print to an external printer (USB, IP printer) through AMP Connect Agent. It can also utilize AMP Connect Agent to perform transactions through AMP payment application. This is suitable for cloud base applications that run on browser.

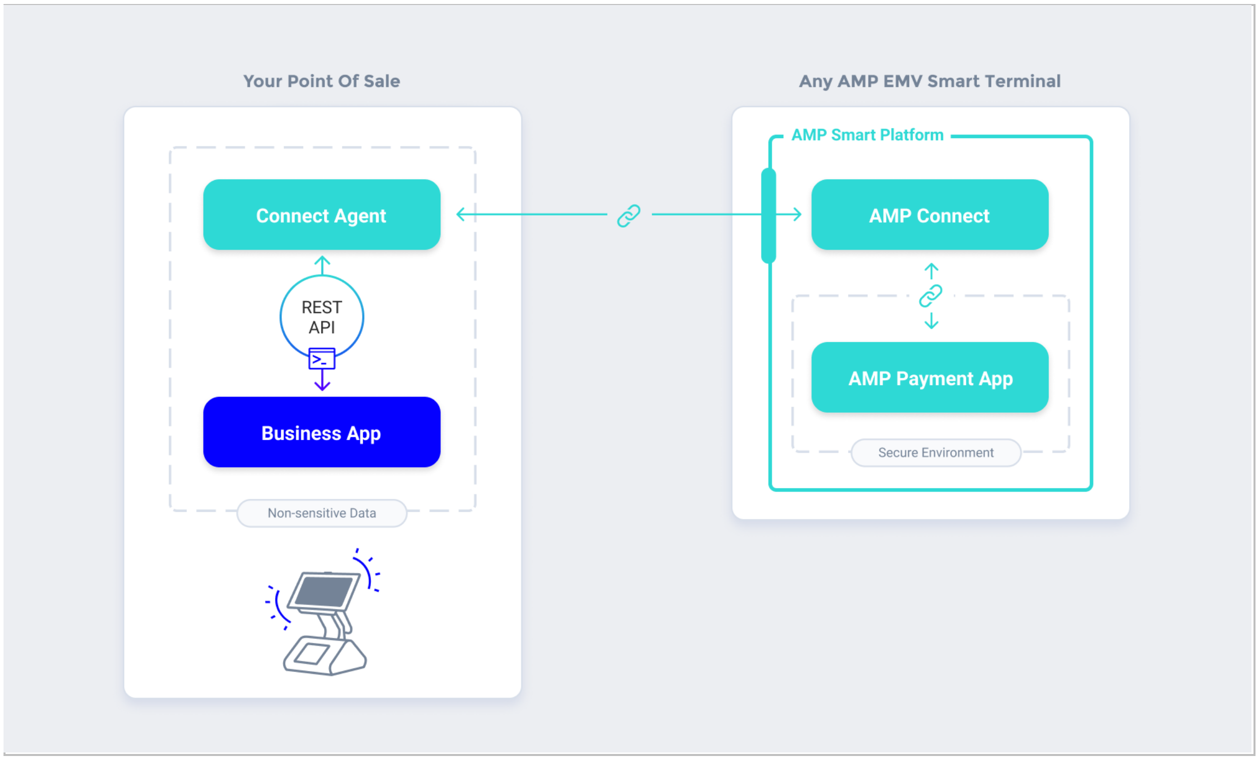


Figure 5 - Formation 3.

3 AMP CONNECT OPERATION MODES

AMP Connect uses different communication media based on business application's requirement. The integration can be wired or wireless in the same network or cloud based. The following diagrams demonstrate how AMP Connect operates with different communication media.

AMP Connect has three modes of operation, which are configurable in ATMS (AMP Terminal Management System):

- USB/Serial
- Wireless in the same network
- Wireless in the cloud

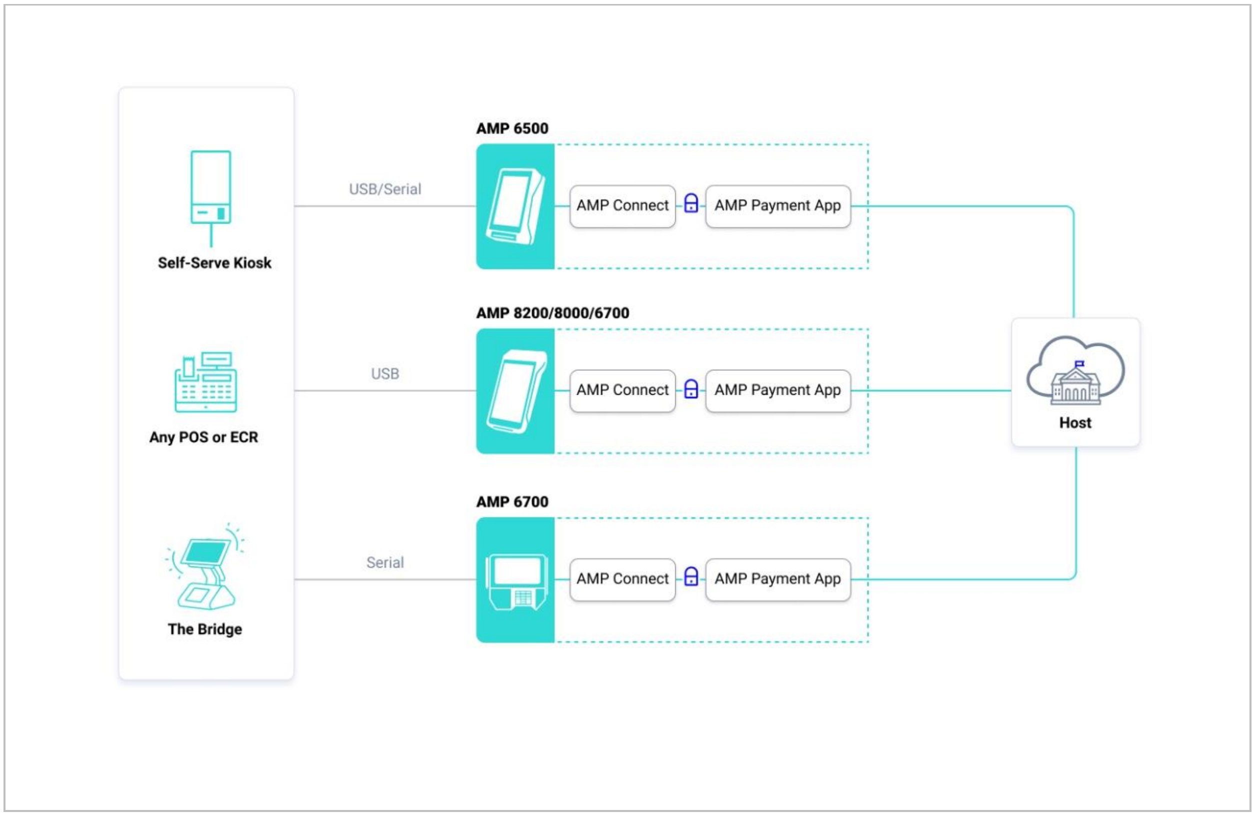


Figure 6 - USB or Serial Integration.

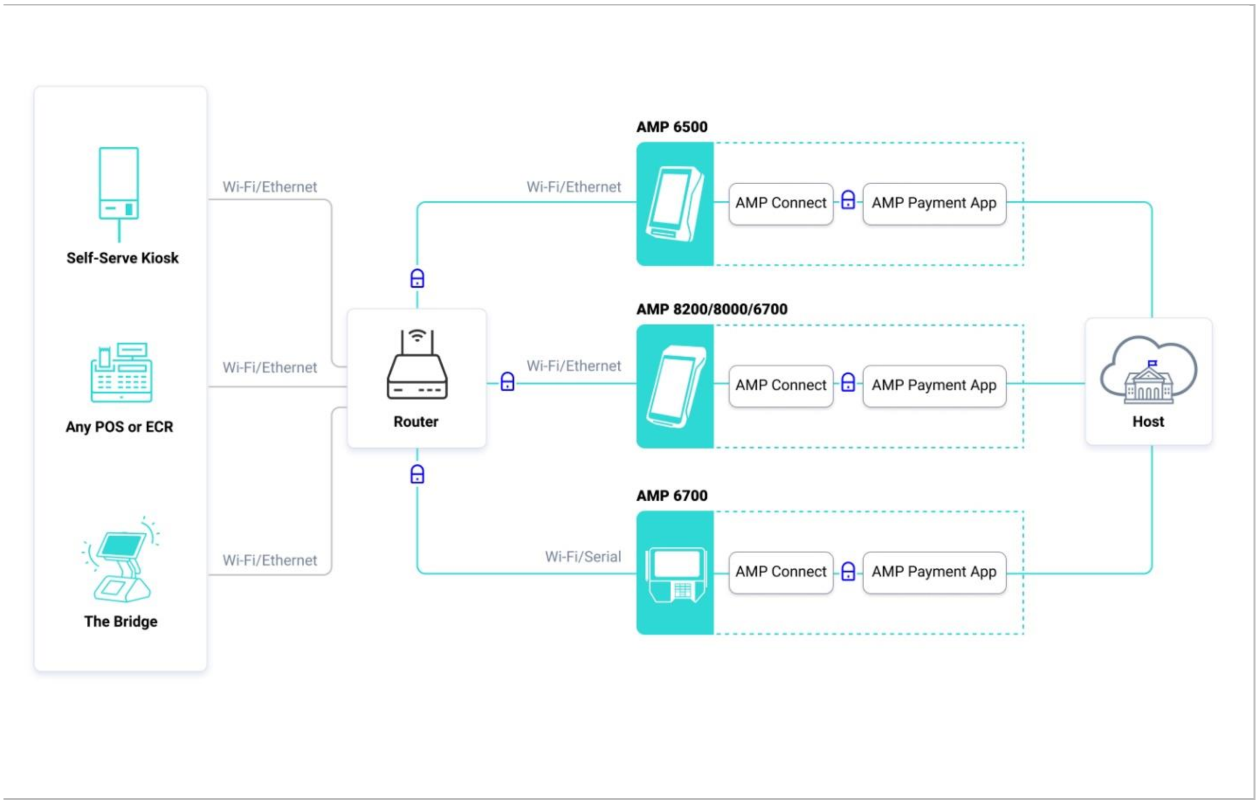


Figure 7 - Same Network Integration through WiFi or Ethernet.

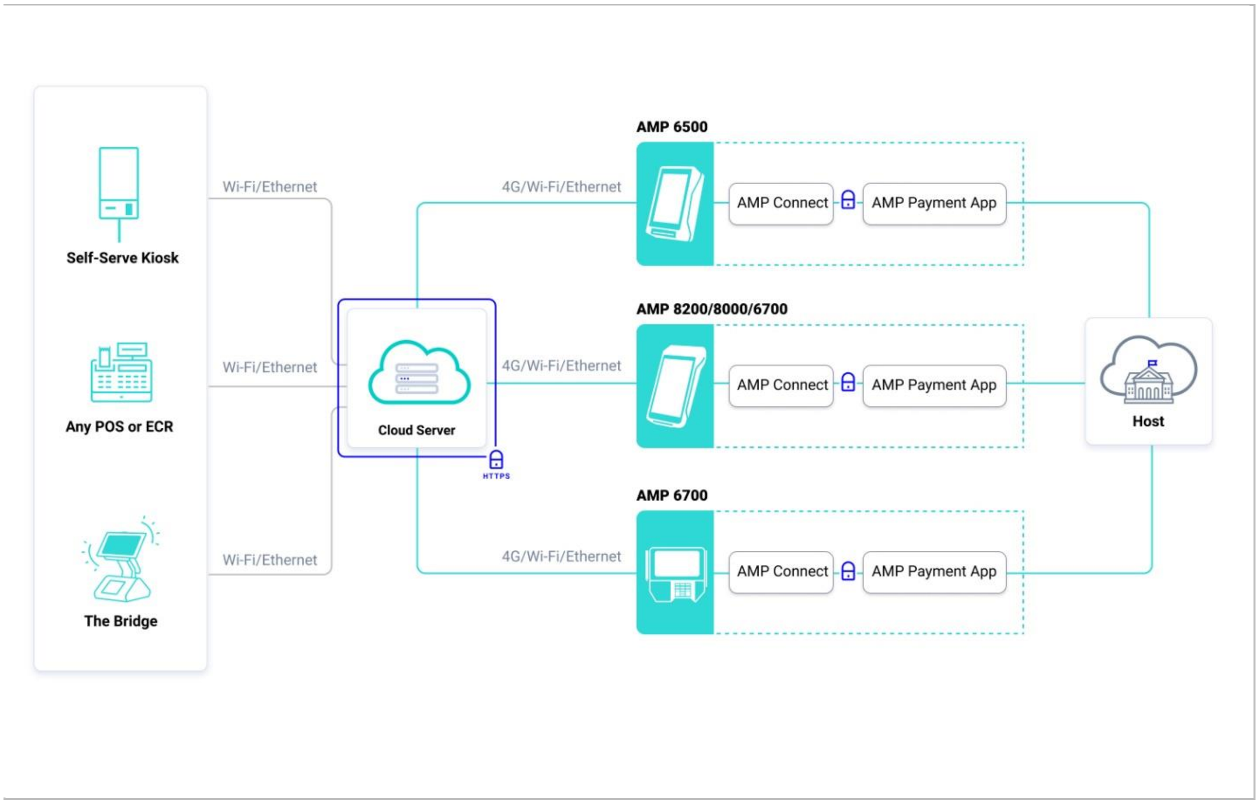


Figure 8 - Cloud Based Integration.

4 USB/SERIAL INTEGRATION

AMP Connect facilitates serial and/or USB connection between a device and AMP terminals. The following cables can be used for this purpose:

Terminal	USB	Serial	Description
AMP8000	AM-CBL-POS-1014	NA	Micro USB to PC USB
AMP8200	AM-CBL-POS-1014	NA	Micro USB to PC USB
AMP6700	AM-CBL-POS-1006		Connect to PC USB for debug and USB connection
		AM-CBL-POS-1016	Connect to PC USB for debug and serial connection
	AM-CBL-POS-1013		Connect to PC USB
AMP6500	AM-CBL-POS-1013		Connect to PC
		AM-CBL-POS-1022	Connect serial port to PC serial port
		AM-CBL-POS-1023	Connect serial to PC USB
		AM-CBL-POS-1024	Connect serial port to a device’s UIB serial

4.1 MESSAGE FORMAT

The message format for USB/Serial connection is described in **Customer Display, Transaction** and **System** sections. Please note that the “endpoint” tag must be added to the message format in USB/Serial mode. The “endpoint” tag value should be set to “CUSTOMER_DISPLAY”, “TRANSACTION” or “SYSTEM”.

- Sample request:

```
{"EndPoint":"TRANSACTION","cmdType":"Purchase","ReqPayload":{“autoPr
int”:”TRUE”,"BaseAmount":"10.00","CardEntryMethod":"AUTO","UserDefin
edEchoData":"123456789"}}
```

- Sample response:

```
{"endPoint":"TRANSACTION","cmdType":"GET_TRANSACTION_RESULT"}
```

Appendix C includes sample messages.

4.2 IMPLEMENTATION PREREQUISITES

1) Load the right template to the terminal from ATMS. AMP informs you of the right template that includes the desired payment application and the AMP Connect app.

2) Make sure the AMP Connect parameter that sets the communication mode is set to USB.

3) Use the right cable to connect the terminal to the device.

4.3 PROGRAMMING STEPS

The “TRANSACTION” and “SYSTEM” endpoint commands are asynchronous. The command is sent to the terminal, the terminal interacts with the customer and then sends back the result. The customer response time and the transaction process determine how long it takes for the result to be sent back. AMP Connect does not interfere with the process and waits until the result is sent back by the terminal.

The business app is responsible for checking the result availability by sending the proper commands.

The following is a simple scenario:

1) Open USB or serial with 8N1, 115200 baud rate.

2) Send a message to serial or USB port

3) Send the “TRANSACTION” endpoint JSON command string.

4) Continue reading the result from USB or serial, setting "cmdType" to "GetTransactionResult", until the result is available.

Appendix D includes detailed samples.

5 IP INTEGRATION ON THE SAME NETWORK

5.1 GETTING THE TERMINAL'S IP ADDRESS

DNS-based Service Discovery (DNS-SD) is the first step that should be performed. If the terminal has static IP address, the following steps are not required.

- The business application requests service by specifying the terminal's serial number as the service name.
- The AMP Connect app registers a service name which matches the terminal's serial number. For instance, if the AMP Connect app is running on an AMP 6700 device with the serial number 6700000100, the DNS-based service discovery should be performed for the exact same name.
- The service type for the DNS-based service discovery should be "_http._tcp."
- The service discovery protocol should be set to "PROTOCOL_DNS_SD"; the DNS based service discovery protocol.
- After a successful service discovery, the client's application can resolve the discovered service and establish a connection for fetching the terminal's IP address.

The sample code for C# and Android can be found in **Appendix B**.

5.2 MESSAGE FORMAT

The message format is described in the **Customer Display**, **Transaction** and **System** sections. Since the endpoint is part of the API URL call, there is no need to add new tags. All commands should be sent as POST.

5.3 GENERATING THE REST API URL ADDRESS

The IP address that is fetched from the DNS-based service discovery is used to create the REST API URL address. The port number is always fixed and is set to "22222". The below pattern should be followed for each REST API call:

```
POST>"http://" + "AMPConnect's IP address" +":22222/" + "endpoint"
```

For instance, when the discovery address is 192.168.0.10, the URL address for a payment related API call should be:

```
“http://192.168.0.10:22222/TRANSACTION”
```

Sample messages are available in **Appendix C**.

5.4 IMPLEMENTATION PREREQUISITES

1) Load the right template to the terminal from ATMS. AMP informs you of the right template that includes the desired payment application and AMP Connect app.
2) Make sure the AMP Connect parameter, that sets the communication mode, is set to WIRELESS.
3) Have the right discovery code in the business application.
4) Start AMP Connect on the terminal first and then start the discovery.
5) Start sending the command from the business app to the endpoint through REST API.

5.5 PROGRAMMING STEPS

The “TRANSACTION” and “SYSTEM” endpoint commands are asynchronous. The command is sent to the terminal, the terminal interacts with the customer and then sends back the result. The customer response time and the transaction process determine how long it takes for the result to be sent back. AMP Connect does not interfere with the process and waits until the result is sent back by the terminal.

The business app is responsible for checking the result availability by sending the proper commands.

The following is a simple scenario:

1) Discover the terminal IP address in network.
2) Call REST API.
3) Continue reading the result, by calling the REST API, setting “cmdType” to “GetTransactionResult”, until result is available.

Appendix D includes detailed samples.

6 CLOUD INTEGRATION

AMP Connect provides the opportunity for a business application to perform transactions without network configuration overhead, by sending the transaction to the AMP Cloud. The AMP

Cloud is hosted by AMP. It accepts the transaction request and routes the transaction to the registered terminals. This method can be used by browser-based applications.

6.1 MESSAGE FORMAT

The message format for Cloud is described in the **Transaction** section. Please note that the “endpoint” tag must be added to message format in Cloud mode. The “endpoint” tag value should be set to “TRANSACTION”.

- Sample request payload:

```
….“payload”:{"endPoint":"TRANSACTION",”autoPrint”:”TRUE”,"baseAmount
":"1.00","cardEntryMethod":"AUTO","echoData":"123456789","cmdType":"
PURCHASE"}
```

- Sample response payload:

```
….“payload”:{"endPoint":"TRANSACTION","cmdType":"GET_TRANSACTION_RES
ULT"}
```

Appendix C includes sample messages.

AMP Connect app resides on the terminal and communicates with the AMP Cloud and BaseApp to perform a transaction.

APIs can be accessed through the below URL:

https://ecr.amobilepayment.com:3000/api/command

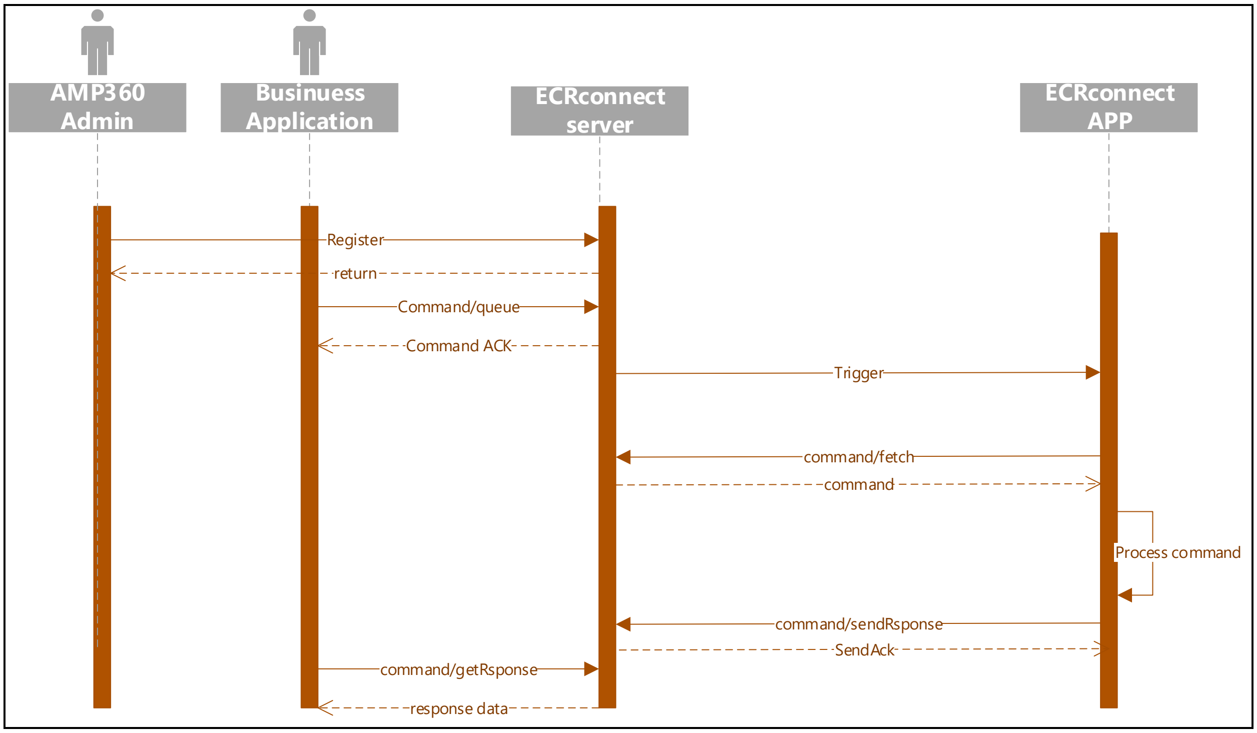


Figure 9 - Cloud Integration Message Flow.

6.1.1 QUEUE COMMAND

The business application should send this command to the AMP Cloud for performing a transaction. The “payload” tag must include transaction specific data.

- Request:

POST/command/queue		
JSON key	**JSON key description**	**Value type**
sourceSerial	An identifier for the device that sends the command. This data is only logged in the server.	String
serialNumber	AMP terminal’s serial number.	String
terminalId	The terminal’s terminal ID in ATMS. This value is not the same as the host TID.	String
authCode	Provided by the AMP Cloud at the time of terminal registration.	String
payload	JSON payload: The command that should be processed by the terminal	JSON

The colored rows identify the tags that are included in the “result” tag.

JSON key	JSON key description	Value type
commandId	The unique command ID that is created by the server, which should be used in future follow up commands.	String
success	Boolean: true or false. Indicating a successful or a failed operation.	Boolean
Message	A message returned by the server to describe the command execution status.	String

6.1.2 STATUS COMMAND

- Request:

GET/command/status/{commandId}		
JSON key	**JSON key description**	**Value type**
commandId	The unique command ID that is returned by the server, in response to a queue command.	String

- Response:

The colored rows identify the tags that are included in the “result” tag.

JSON key	JSON key description	Value type
commandId	Command ID	String
sourceSerial	The identifier for the device that sent the command, which was originally sent in the request.	String
targetTerminal	The serial number of the target terminal that processed the request.	String
status	Command status: INITIALIZED, SENT_PAYLOAD, RECEIVED_RESPONSE, DONE	String
requestData	The requested payload data.	String
responseData	Data received from the terminal. Can be NULL.	String

createdAt	The command creation timestamp, created by the server.	String
updatedAt	The command update timestamp, created by the server.	String
TerminalId	The terminal's terminal ID in ATMS that was sent originally in the request. This value is not the same as the host TID.	String
canReload	Boolean: true or false. Indicating a successful or a failed operation.	Boolean
message	A message returned by the server to describe the command execution status.	String

6.1.3 RESPONSE COMMAND

When the command status is either RECEIVED_RESPONSE or DONE, this command returns the actual response from the terminal. Otherwise, it returns the status. The business app should continue calling this command until it detects the status value RECEIVED_RESPONSE or DONE, which means response data is available.

- Request

GET/command/response/{commandId}		
JSON key	**JSON key description**	**Value type**
commandId	The unique command ID that is returned by the server, in response to a queue command.	String

- Response:

The colored rows identify the tags that are included in the "result" tag.

JSON key	**JSON key description**	**Value type**
canReload	Boolean: true or false. Indicating a successful or a failed operation.	Boolean
status	Command status: INITIALIZED, SENT_PAYLOAD, RECEIVED_RESPONSE, DONE	String

Message	A message returned by the server to describe the command execution status.	String
responseData	The returned data after transaction execution by the terminal. A null value means no response is received from the terminal.	JSON

6.1.4 FLUSH COMMAND

This command is exclusively between the business application and the AMP Cloud.

This command is not queued and is processed immediately.

The Flush command clears all the commands from the command queue and sets the status of all pending commands to DONE.

- Request:

POST/command/flush		
JSON key	**JSON key description**	**Value type**
terminalId	The terminal's serial number	String
authCode	Provided by the AMP Cloud at the time of terminal registration.	String

- Response

JSON key	**JSON key description**	**Value type**
canReload	Boolean: true or false. Indicating a successful or a failed operation.	Boolean
Message	A message returned by the server to describe the command execution status.	String

6.2 IMPLEMENTATION PREREQUISITES

1) Register the terminal in the AMP Cloud to generate the auth code.

2) Make sure the AUTHCODE parameter in ATMS is set to the auth code value that is generated during terminal registration.

3) Make sure the AMP Connect parameter, that sets the communication mode, is set to CLOUD.

4) Load the right template to the terminal from ATMS. AMP informs you of the right template that includes the desired payment application and AMP Connect app.

5) Start sending the command from the business app to the AMP Cloud.

6.3 PROGRAMMING STEPS

The "TRANSACTION" endpoint commands are asynchronous. The command is sent to the terminal, the terminal interacts with the customer and then sends back the result. The customer response time and the transaction process determine how long it takes for the result to be sent back. AMP Connect does not interfere with the process and waits until the result is sent back by the terminal.

The business app is responsible for checking the result by sending the proper commands.

The following is a simple scenario:

1) Call the Queue command.

2) Continue reading the result, by sending the Response command, until the result is available. Result is available when the "status" tag value is DONE or RECEIVED_RESPONSE.

Appendix D includes detailed samples.

7 DISPLAY

The business app uses this API to customize the AMP Connect app screen or display an object on the terminal's screen.

The following URL should be used for Display API calls:

Endpoint URL: DISPLAY.

Sample for WIFI: http://IP_Address:22222/DISPLAY

The following sections describe common Display API calls.

7.1 DISPLAY REQUEST

The AMP Connect data request is in JSON format. There are two mandatory JSON keys: "cmdType" and "ReqPayload". Command parameters are defined inside the "ReqPayload" JSON object.

Command Request			
Request Format	`{“cmdType”:”, “ReqPayload”:{“”:””, “”:””,“”:””, ……}}`		
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: calling the Display functions.	String	Display ClearDisplayText IdleScreen DisplayQR ClearQR
ReqPayload	The request payload, which defines Display data.	String (JSON Object)	

7.2 DISPLAY RESPONSE

The response from the AMP Connect app only indicates whether the command is successfully received and is properly formatted. It does not include any details about command execution. An example of the AMP Connect response code is:

```
{“AMPConnectResponseCode”: “0”}
```

7.3 DISPLAY COMMANDS

7.3.1 DISPLAYTEXT

This method sends HTML data to be displayed on the terminal’s screen.

Prototype	`“cmdType”:”DisplayText”`
Mandatory Fields	**DisplayData**
Optional Fields	
Application Type	Attended, Unattended
Notes	The DisplayData is in HTML format that will be displayed on the terminal’s screen by the AMP Connect app. **Note**: The double quote is a keyword in JSON. Therefore, a **“** in HTML data should be replaced by **\”.**

- Example:

```
{"cmdType":"DisplayText","ReqPayload":{"DisplayData":"<!DOCTYPE
html>
<html>
<body>
<h1 style=\"color:red\">My First Heading</h1>
 <IMG
SRC=\"https://www.sheldonbrown.com/images/scb_eagle_contact.jpeg\">
<CENTER>
      <p>My first paragraph</p>
</CENTER>
</body>
</html>"}}
```

7.3.2 CLEARDISPLAYTEXT

This method clears the display and shows the idle screen.

Prototype	`“cmdType”:”ClearDisplayText”`
Mandatory Fields	
Optional Fields	
Application Type	Attended, Unattended
Notes	

- Example:

```
{"cmdType":"ClearDisplayText","ReqPayload":{}}
```

7.3.3 IDLESCREEN

This method sets the AMP Connect app’s idle screen.

Prototype	`“cmdType”:”IdleScreen”`
Mandatory Fields	**DisplayData** **DefaultIdleScreen**
Optional Fields	
Application Type	Attended, Unattended

Notes	When DefaultIdleScreen is set to “true”, the default AMP Connect app’s Idle screen is displayed. **Note:** the display data is an image that should be converted to base64 and should be the same size as the screen. The acceptable image formats are bitmap, PNG, TIF, etc.

- An example for the set Idle screen:

```
{"cmdType":"IdleScreen","ReqPayload":{"DisplayData":" BASE24 of Image file”,"DefaultIdleScreen":"false"}}
```

- An example for the default idle screen:

```
{"cmdType":"IdleScreen","ReqPayload":{"DisplayData":"",
"DefaultIdleScreen":"true"}}
```

7.3.4 DISPLAYQR

This method shows the QR code on the terminal’s screen. The QR code is only displayed when the idle screen is displayed.

Prototype	`“cmdType”:”DisplayQR”`
Mandatory Fields	**DisplayData**
Optional Fields	
Application Type	Attended, Unattended
Notes	

- Example:

```
{"cmdType":"DisplayQR","ReqPayload":{"DisplayData":"This is QR to display"}}
```

7.3.5 CLEARQR

This method clears the display and shows the idle screen.

Prototype	`“cmdType”:”ClearQR”`
Mandatory Fields	
Optional Fields	
Application Type	Attended, Unattended
Notes	

- Example:

```
{"cmdType":"ClearQR","ReqPayload":{}}
```

8 CAPTURE

The business app uses this API to create a customized data collection screen, e.g. email entry.

The following URL should be used for all Capture API calls:

Endpoint URL: CAPTURE.

Sample for WIFI: http://IP_Address:22222/CAPTURE

The following sections describe the common Capture API calls.

8.1 CAPTURE REQUEST

The AMP Connect data request is in JSON format. There are two mandatory JSON keys: “cmdType” and “ReqPayload”. The command parameters are defined inside the “ReqPayload” JSON object.

Command Request			
Request Format	`{“cmdType”:”, “ReqPayload”:{“”:””, “”:””,“”:””, ……}}`		
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action.	String	CaptureSignature CaptureEmail CaptureSurvey CaptureQR GetCaptureResult

ReqPayload	The request payload, which defines the Capture data	String (JSON Object)	

8.2 CAPTURE RESPONSE

8.2.1 AMP CONNECT RESPONSE CODE

The response from the AMP Connect app only indicates whether the command is successfully received and is properly formatted. It does not include any details about command execution. An example of the AMP Connect response code is:

```
{“AMPConnectResponseCode”: “0”}
```

To get the captured data, use the GetCaptureResult command. This command can only be used in WLAN and USB modes. In cloud mode, the captured data is returned to the cloud. The data then can be fetched, using cloud commands.

8.3 CAPTURE COMMANDS

8.3.1 CAPTURESIGNATURE

This method instructs the AMP Connect app to collect a customer’s signature. The data is returned when the signature is captured and approved by the customer.

Prototype	`“cmdType”:”CaptureSignature”`
Mandatory Fields	
Optional Fields	
Application Type	Attended, Unattended
Notes	The signature data is captured in base64 format of the signature bitmap.

- Example:

```
{"cmdType":"CaptureSignature","ReqPayload":{}}
```

8.3.2 CAPTUREEMAIL

This method instructs the AMP Connect app to collect a customer’s email. The data is returned when the email is captured and approved by the customer.

Prototype	`“cmdType”:” CaptureEmail”`
Mandatory Fields	
Optional Field	
Application Type	Attended, Unattended
Notes	

- Example:

```
{"cmdType":"CaptureEmail","ReqPayload":{}}
```

8.3.3 CAPTUREQR

This method instructs the AMP Connect app to read the barcode by turning on the camera.

Prototype	`“cmdType”:” CaptureQR”`
Mandatory Fields	
Optional Fields	
Application Type	Attended, Unattended
Notes	

- Example:

```
{"cmdType":"CaptureQR","ReqPayload":{}}
```

8.3.4 CAPTURESURVEY

This method instructs the AMP Connect app to show an HTML page that contains data entry fields. The collected data is returned when data is captured and submitted by the customer.

Prototype	`“cmdType”:” CaptureSurvey”`
Mandatory Fields	SurveyData
Optional Fields	
Application Type	Attended, Unattended

Notes	The data entry screen, or HTML data, should be designed to completely cover the screen. This screen must have both 'Submit' and 'Cancel' buttons. The 'Submit' button uses the AMPConnect.submitSurvey("User data") JavaScript command and the 'Cancel' button uses the AMPConnect.cancelSurvey() JavaScript command. When 'submitSurvey' is called, the AMP Connect app captures the entered data and returns it as part of the result. The captured data is a string with "," separator per collected filed.

- Example:

```
{"cmdType":"CaptureSurvey","ReqPayload":{"SurveyData":"<!DOCTYPE html>
<html>
<body>
  <script>
    function submitSurvey() {
        var firstName = document.getElementById(\"fname\").value;
        var lastName = document.getElementById(\"lname\").value;
             var checkBox1 = document.getElementById(\"option1\").checked;
             var checkBox2 = document.getElementById(\"option2\").checked;
        AMPConnect.submitSurvey(firstName + \",\" + lastName + \",\" +
checkBox1 + \",\" + checkBox2);
    }
    function cancelSurvey() {
        AMPConnect.cancelSurvey();
    }
  </script>
  <h1>Survey</h1>
  <label>First name:</label>
  <input type=\"text\" id=\"fname\" name=\"fname\"><br><br>
  <label>Last name:</label>
  <input type=\"text\" id=\"lname\" name=\"lname\"><br><br>
  <input type=\"checkbox\" id=\"option1\" name=\"option1\" value=\"option1
value\">
  <label> Option1</label><br>
  <input type=\"checkbox\" id=\"option2\" name=\"option2\" value=\"option2
value\">
  <label> Option2</label><br><br>
  <input type=\"submit\" value=\"Cancel\" onclick=\"cancelSurvey()\">
  <input type=\"submit\" value=\"Submit\" onclick=\"submitSurvey()\">
</body>
</html>"}}
```

8.3.5 GETCAPTURERESULT

The Capture endpoint requires customer interaction. Therefore, the response is asynchronous. This method returns the data that is entered by the customer. This command should be executed in a loop until a response is received, or timeout is reached. The timeout is handled by the business app. In Cloud mode, this command is not required, because AMP Connect sends the result to the cloud.

Prototype	`"cmdType":" GetCaptureResult"`
Mandatory Fields	
Optional Fields	

Response	**Payload**: Contains the captured data
	ResponseCode:
	CAPTURE_COMPLETE = 23;
	CAPTURE_CANCELLED = 24;
	CAPTURE_TIMEOUT = 25;

Application Type	Attended, Unattended
Notes	

Example:

```
{"cmdType":"GetCaptureResult","ReqPayload":{}}
```

9 CUSTOMER DISPLAY

This section addresses the Customer Display API calls, including the JSON message protocol for sending data to the AMP Connect app, the sent data types and the corresponding command actions. For example, adding items for purchase, voiding items, cash tendered, clearing the screen and copying product images to the POS/ECR device so that the correct image is displayed for each request.

Note that customer display is exclusive to AMP 6700.

The following URL should be used for all customer display API calls:

Endpoint: CUSTOMER_DISPLAY

Sample for WiFi: http://IP_Address:22222/CUSTOMER_DISPLAY

9.1 CUSTOMER DISPLAY COMMANDS

9.1.1 ADDING ITEMS TO THE LIST

This command is used when the customer purchases a selected item. The following describes the command format:

Adding items to the list			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: adding an item to the list.	String	"ADD_ITEM"
amount	The added item's price.	String	e.g. "3.99"
total	The total cost of all items, including the tax value.	String	e.g. "15.48"
quantity	The total quantity of the added items.	Integer	e.g. 4
item_type	For adding an item, this value should be set to "NORMAL". For voiding an item, this value should be set to "VOID". For refund, this value should be set to "REFUND".	String	"NORMAL" "VOID" "REFUND"
subtotal	The total cost of the added items, excluding taxes.	String	e.g. "14.37"
Item_number	The added item's unique identifier within the folder which will then be loaded to the terminal. This identifier can be used if the merchandise items' icons are stored in the device's internal storage.	Integer	e.g. 27
description	The added item's name.	String	e.g. "Coca cola 330 ml"

tax	The tax value that is added to the purchase.	String	e.g. “2.12”

- Sample ADD_ITEM Request and Response
  - Request:

```
http://10.242.3.90:22222/CUSTOMER_DISPLAY
{"amount":3.99,"description":"Coca
cola","item_number":123,"item_type":"NORMAL","quantity":3,"subtotal"
:9.99,"tax":1.00,"total":10.99,"cmdType":"ADD_ITEM"}
```

  - Response:

```
{“AMPConnectResponseCode”:0}
```

9.1.2 CLOSING THE RECEIPT

Closing the receipt indicates that all the actions are done. Hence, the total can be displayed to the customer on the POS/ECR screen. The following describes the command format:

Closing the receipt			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: closing the receipt.	String	“CLOSE_RECEIPT”
total	The total cost at the time of receipt closure.	String	e.g. “76.32”

- Sample CLOSE_RECEIPT Request and Response
  - Request:

```
http://10.242.3.90:22222/CUSTOMER_DISPLAY
{"total":36.89,"cmdType":"CLOSE_RECEIPT"}
```

  - Response

```
{“AMPConnectResponseCode”:0}
```

9.1.3 CASH TENDERED

This command specifies the tendered amount, which is the amount of money provided by the customer for the purchase. This value should be greater than or equal to the purchase total. The following describes the command format:

The tendered amount			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: displaying the tendered amount by customer.	String	“CASH_TENDERED”
tenderedAmount	The tendered amount, which is the amount of money provided by the customer for the purchase. This value should be greater than or equal to the purchase total.	String	e.g. “50.00”
due	The money due to the customer, which is the difference between the tendered amount and the total.	String	e.g. “17.49”

- Sample CASH_TENDERED Request and Response
  - Request:

```
http://10.242.3.90:22222/CUSTOMER_DISPLAY
{"due":17.49,"tenderedAmount":50.00,"cmdType":"CASH_TENDERED"}
```

  - Response:

```
{“AMPConnectResponseCode”:0}
```

9.1.4 CLEARING THE SCREEN

This action removes all the previously added items. Additionally, it clears the subtotal, tax, and total values, providing the user with a fresh start. The following describes the command format:

The tendered amount			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: clearing the items list and providing the user with a fresh start.	String	“CLEAR_SCREEN”

- Sample CLEAR_SCREEN Request and Response
  - Request:

```
http://10.242.3.90:22222/CUSTOMER_DISPLAY
{"cmdType":"CLEAR_SCREEN"}
```

  - Response:

```
{“AMPConnectResponseCode”:0}
```

10 TRANSACTION

The primary purpose of this API is to facilitate the integration between the AMP Connect app and the AMP payment application, BaseApp, which supports Credit/Debit payments or alternative payment methods like Alipay/WeChatPay.

The following URL should be used for all payment API calls:

Endpoint URL: TRANSACTION.

Sample for WiFi: http://IP_Address:22222/TRANSACTION

The following sections describe the common transaction API calls that can be triggered from another device for communicating with the AMP payment application.

10.1 TRANSACTION REQUESTS

The AMP Connect data request is in JSON format. There are two mandatory JSON keys: “cmdType” and “ReqPayload”. The parameters for each transaction type are defined inside the “ReqPayload” JSON object.

Transaction Request			
Request Format	`{“cmdType”:”, “ReqPayload”:{“”:””, “”:””,“”:””, ……}}`		
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: calling a payment transaction.	String	See **Transaction Commands** for details.
ReqPayload	The request payload, which defines all transaction parameters	String (JSON Object)	See **Transaction Input Tags** for details.

10.2TRANSACTION RESULT

To get the payment transaction result, call the “GetTransactionResult” command, which retrieves the last transaction’s data. The response data format is unified across all payment transaction commands. The following table includes the details.

JSON key	JSON key description	Value
CommandType	The command type. Currently, this value is always set to “TRANSACTION”.	“TRANSACTION”
Command	This value should be the same as the ‘cmdType” in the request.	See **Transaction Commands** for details.
Status	The payment transaction’s status.	“RESULT_OK” “RESULT_CANCELLED”
UserDefinedEchoData	This value matches the data in the request.	
Payload	The actual payment transaction response data that is returned by the AMP payment app.	

10.3 TRANSACTION COMMANDS

10.3.1 PURCHASE

This method calls BaseApp's Purchase / Sale transaction.

Prototype	`"cmdType":"Purchase"`
Mandatory Fields (See **Transaction Input Tags** for details)	Auto-Print Card Entry Method Card Number (PAN)* Expiry Date* Card Security Code/ Card Verification Value* Base Amount**
Optional Fields (See **Transaction Input Tags** for details)	Tip Amount Address Echo Data Cashback Amount Tax Amount Additional Data
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL

	KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_CASHBKAMT KEY_RESPONSE_SURCHARGEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT

	KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_TOKEN_DATA KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_CASHBACKFEE KEY_RESPONSE_NONCASHFEE KEY_RESPONSE_EMV_SETTLEDATE KEY_RESPONSE_EMV_NETWORKID KEY_RESPONSE_EMV_APPLABEL*** KEY_RESPONSE_EMV_APPLPAN*** KEY_RESPONSE_EMV_CURRENCYCODE*** KEY_RESPONSE_EMV_PANSEQNUM*** KEY_RESPONSE_EMV_AIP***

	KEY_RESPONSE_EMV_TVR*** KEY_RESPONSE_EMV_TRANSDATE*** KEY_RESPONSE_EMV_TRANSTYPE*** KEY_RESPONSE_EMV_AMT_AUTH*** KEY_RESPONSE_EMV_OTH_AMT*** KEY_RESPONSE_APP_USAGE_CTRL*** KEY_RESPONSE_EMV_IAC_ONLINE*** KEY_RESPONSE_EMV_IAC_DEFAULT*** KEY_RESPONSE_EMV_IAC_DENIAL*** KEY_RESPONSE_APPPREFNAME*** KEY_RESPONSE_EMV_TERM_CCODE*** KEY_RESPONSE_EMV_AC*** KEY_RESPONSE_EMV_CID*** KEY_RESPONSE_EMV_IAD*** KEY_RESPONSE_EMV_CVM*** KEY_RESPONSE_EMV_ATC*** KEY_RESPONSE_EMV_UNPRED_NUM*** KEY_RESPONSE_EMV_TAC_ONLINE*** KEY_RESPONSE_EMV_TAC_DEFAULT*** KEY_RESPONSE_EMV_TAC_DENIAL*** KEY_RESPONSE_REVERSAL_AMT KEY_RESPONSE_APP_VERSION KEY_RESPONSE_EMV_APP_TRANSDATA KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_SWVER **** KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID *****

	KEY_RESPONSE_BANKRESPONSECODE ***** KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Attended, Unattended
Notes	The following response fields only contain value when the Re-enter transaction is executed. This transaction is exclusively supported by TSYS: KEY_RESPONSE_EMV_SETTLEDATE KEY_RESPONSE_EMV_NETWORKID

10.3.2 FORCE POST

This method calls BaseApp's Force Post transaction.

Prototype	`“cmdType”:”ForcePost”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Card Entry Method** **Card Number (PAN)*** **Expiry Date*** **Card Security Code/ Card Verification Value*** **Authorization Code*** **Transaction** ID***** **Base Amount**
Optional Fields (See **Transaction Input Tags** for details)	**Address** **Zip Code** **Echo Data** **Additional Data**
Key Response	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1

(refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR **** KEY_RESPONSE_AID **** KEY_RESPONSE_AUTHCODE **** KEY_RESPONSE_SEQUENCENUM **** KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD **** KEY_RESPONSE_TSI ****

	KEY_RESPONSE_ARC **** KEY_RESPONSE_CODE **** KEY_RESPONSE_TEXT **** KEY_RESPONSE_APPPREFNAME **** KEY_RESPONSE_TC **** KEY_RESPONSE_HOSTTIMESTAMP **** KEY_RESPONSE_TRANSREF **** KEY_RESPONSE_RESULTCODE KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_APP_VERSION KEY_RESPONSE_REVERSAL_AMT **** KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_SWVER **** KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID ***** KEY_RESPONSE_BANKRESPONSECODE *****

	KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Attended

10.3.3 REFUND

This method calls BaseApp's Refund transaction.

Prototype	`"cmdType":"Refund"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Card Entry Method** **Card Number (PAN)*** **Expiry Date*** **Card Security Code/ Card Verification Value*** **Base Amount**
Optional Fields (See **Transaction Input Tags** for details)	**Address** **Zip Code** **Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL

	KEY_RESPONSE_PAN KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF

	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_EMV_SETTLEDATE KEY_RESPONSE_EMV_NETWORKID KEY_RESPONSE_EMV_APPLABEL*** KEY_RESPONSE_EMV_APPLPAN*** KEY_RESPONSE_EMV_CURRENCYCODE*** KEY_RESPONSE_EMV_PANSEQNUM*** KEY_RESPONSE_EMV_AIP*** KEY_RESPONSE_EMV_TVR*** KEY_RESPONSE_EMV_TRANSDATE*** KEY_RESPONSE_EMV_TRANSTYPE*** KEY_RESPONSE_EMV_AMT_AUTH*** KEY_RESPONSE_EMV_OTH_AMT*** KEY_RESPONSE_APP_USAGE_CTRL***

	KEY_RESPONSE_EMV_IAC_ONLINE*** KEY_RESPONSE_EMV_IAC_DEFAULT*** KEY_RESPONSE_EMV_IAC_DENIAL*** KEY_RESPONSE_APPPREFNAME*** KEY_RESPONSE_EMV_TERM_CCODE*** KEY_RESPONSE_EMV_AC*** KEY_RESPONSE_EMV_CID*** KEY_RESPONSE_EMV_IAD*** KEY_RESPONSE_EMV_CVM*** KEY_RESPONSE_EMV_ATC*** KEY_RESPONSE_EMV_UNPRED_NUM*** KEY_RESPONSE_EMV_TAC_ONLINE*** KEY_RESPONSE_EMV_TAC_DEFAULT*** KEY_RESPONSE_EMV_TAC_DENIAL*** KEY_RESPONSE_REVERSAL_AMT KEY_RESPONSE_APP_VERSION KEY_RESPONSE_EMV_APP_TRANSDATA *** KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_SWVER **** KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID ***** KEY_RESPONSE_BANKRESPONSECODE ***** KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Attended

10.3.4 VOID

This method calls BaseApp's Void transaction. Unattended applications always void the last transaction and therefore **Void Type** and **Void Number** are not required.

Prototype	`”cmdType”:”Void”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Void Type**** **Void Number**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM

	KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_CASHBKAMT KEY_RESPONSE_SURCHARGEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_TOKEN_DATA KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3

	KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_CASHBACKFEE KEY_RESPONSE_NONCASHFEE KEY_RESPONSE_EMV_SETTLEDATE KEY_RESPONSE_EMV_NETWORKID KEY_RESPONSE_APP_VERSION KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_EMV_APPLABEL*** KEY_RESPONSE_EMV_APPLPAN*** KEY_RESPONSE_EMV_CURRENCYCODE*** KEY_RESPONSE_EMV_PANSEQNUM*** KEY_RESPONSE_EMV_AIP*** KEY_RESPONSE_EMV_TVR*** KEY_RESPONSE_EMV_TRANSDATE*** KEY_RESPONSE_EMV_TRANSTYPE*** KEY_RESPONSE_EMV_AMT_AUTH*** KEY_RESPONSE_EMV_OTH_AMT*** KEY_RESPONSE_APP_USAGE_CTRL***

	KEY_RESPONSE_EMV_IAC_ONLINE*** KEY_RESPONSE_EMV_IAC_DEFAULT*** KEY_RESPONSE_EMV_IAC_DENIAL*** KEY_RESPONSE_APPPREFNAME*** KEY_RESPONSE_EMV_TERM_CCODE*** KEY_RESPONSE_EMV_AC*** KEY_RESPONSE_EMV_CID*** KEY_RESPONSE_EMV_IAD*** KEY_RESPONSE_EMV_CVM*** KEY_RESPONSE_EMV_ATC*** KEY_RESPONSE_EMV_UNPRED_NUM*** KEY_RESPONSE_EMV_TAC_ONLINE*** KEY_RESPONSE_EMV_TAC_DEFAULT*** KEY_RESPONSE_EMV_TAC_DENIAL*** KEY_RESPONSE_REVERSAL_AMT KEY_RESPONSE_APP_VERSION KEY_RESPONSE_EMV_APP_TRANSDATA *** KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_SWVER **** KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID ***** KEY_RESPONSE_BANKRESPONSECODE ***** KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Attended, Unattended

10.3.5 PRE-AUTHORIZATION

This method calls BaseApp's Pre-Authorization transaction.

Prototype	`"cmdType":"PreAuth"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Card Entry Method** **Expiry Date*** **Card Security Code/ Card Verification Value*** **Base Amount****
Optional Fields (See **Transaction Input Tags** for details)	**Tip Amount** **Address** **Zip Code** **Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_TRACK1

	KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_CASHBKAMT KEY_RESPONSE_SURCHARGEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP

	KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_TOKEN_DATA KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_CASHBACKFEE KEY_RESPONSE_NONCASHFEE KEY_RESPONSE_EMV_APPLABEL*** KEY_RESPONSE_EMV_APPLPAN*** KEY_RESPONSE_EMV_CURRENCYCODE*** KEY_RESPONSE_EMV_PANSEQNUM*** KEY_RESPONSE_EMV_AIP*** KEY_RESPONSE_EMV_TVR*** KEY_RESPONSE_EMV_TRANSDATE*** KEY_RESPONSE_EMV_TRANSTYPE*** KEY_RESPONSE_EMV_AMT_AUTH***

	KEY_RESPONSE_EMV_OTH_AMT*** KEY_RESPONSE_APP_USAGE_CTRL*** KEY_RESPONSE_EMV_IAC_ONLINE*** KEY_RESPONSE_EMV_IAC_DEFAULT*** KEY_RESPONSE_EMV_IAC_DENIAL*** KEY_RESPONSE_APPPREFNAME*** KEY_RESPONSE_EMV_TERM_CCODE*** KEY_RESPONSE_EMV_AC*** KEY_RESPONSE_EMV_CID*** KEY_RESPONSE_EMV_IAD*** KEY_RESPONSE_EMV_CVM*** KEY_RESPONSE_EMV_ATC*** KEY_RESPONSE_EMV_UNPRED_NUM*** KEY_RESPONSE_EMV_TAC_ONLINE*** KEY_RESPONSE_EMV_TAC_DEFAULT*** KEY_RESPONSE_EMV_TAC_DENIAL*** KEY_RESPONSE_REVERSAL_AMT KEY_RESPONSE_APP_VERSION KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_EMV_APP_TRANSDATA *** KEY_RESPONSE_SWVER KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID ***** KEY_RESPONSE_BANKRESPONSECODE ***** KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Attended, Unattended

10.3.6 COMPLETION

This method calls BaseApp's Completion transaction. Unattended applications allow completion by Invoice Number, Trace Number, Auth Number, Reference Number, Gateway ID and Last Transaction.

Prototype	`“cmdType”:”Completion”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Base Amount**** **Completion Number** **Completion Type****
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE

	KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_CASHBKAMT KEY_RESPONSE_SURCHARGEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE

	KEY_RESPONSE_TOKEN_DATA KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_CASHBACKFEE KEY_RESPONSE_NONCASHFEE KEY_RESPONSE_APP_VERSION KEY_RESPONSE_REVERSAL_AMT **** KEY_RESPONSE_EMV_APP_TRANSDATA **** KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_SWVER **** KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID ***** KEY_RESPONSE_BANKRESPONSECODE ***** KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Attended, Unattended

10.3.7 PRE-AUTHORIZATION CANCELLATION

This method calls BaseApp's Pre-Authorization Cancellation (Completion by zero amount) transaction. This command is only applicable to unattended applications.

Prototype	`"cmdType":"PreAuthCancel"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID

	KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_CASHBKAMT KEY_RESPONSE_SURCHARGEAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_APP_VERSION KEY_RESPONSE_REVERSAL_AMT **** KEY_RESPONSE_EMV_APP_TRANSDATA ****

	KEY_RESPONSE_CODE_ISO **** KEY_RESPONSE_SWVER **** KEY_RESPONSE_ISSUERNAME ***** KEY_RESPONSE_GATEWAYTRANSID ***** KEY_RESPONSE_BANKRESPONSECODE ***** KEY_RESPONSE_AVAILABLE_AMT *****
Application Type	Unattended

10.3.8 SETTLEMENT

This method calls BaseApp's Settlement transaction.

Prototype	`“cmdType”:”Settlement”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The XML format of KEY_RESPONSE_XMLDATA varies per processor.

- The following format applies to the TSYS and GlobalPay processors:

```
<XMLResponse>
    <XMLCmdType>Settlement</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
       <ReportLabel>SETTLEMENT REPORT</ReportLabel>
```

```
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
           <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:54:25</TransTime>
            <MerchantID>887000002726</MerchantID>
            <TerminalID>71134104</TerminalID>
             <FromDate>03/21/2020</FromDate>
             <FromTime>05:31 AM</FromTime>
             <ToDate>03/21/2020</ToDate>
             <ToTime>05:31 AM</ToTime>
             <BatchNo>238</BatchNo>
        </ReportHeader>
        <ReportBody>
            <Transaction_1>
                <InvoiceNo>0000009</InvoiceNo>
                <TransactionType>SL</ TransactionType>
                <AccountNumber>0057</AccountNumber>
                <AuthNumber>552442</AuthNumber>
                <TotalAmount>10.00</TotalAmount>
            </Transaction_1>
            <TransCount>1</TransCount>
        </ReportBody>
        <TerminalTotals>
            <BatchTotalRecords>2</BatchTotalRecords>
            <TotalVoidCount>0</TotalVoidCount>
            <NetSales>10.00</NetSales>
        </TerminalTotals>
    </XMLReceipt>
</XMLResponse>
```

- The following format applies to the PayFacto processor:

```
<XMLResponse>
    <XMLCmdType>Transaction</XMLCmdType>
    <XMLBody>
        <TransType>Settlement</TransType>
    </XMLBody>
    <XMLReceipt>
        <ReportHeader>
            <ReceiptHeader1>NONE</ReceiptHeader1>
            <ReceiptHeader2>NONE</ReceiptHeader2>
            <ReceiptHeader3>NONE</ReceiptHeader3>
            <ReceiptHeader4>NONE</ReceiptHeader4>
            <ReceiptHeader5>NONE</ReceiptHeader5>
            <ReceiptHeader6>NONE</ReceiptHeader6>
            <TransDate>04/22/19</TransDate>
            <TransTime>06:43:37</TransTime>
            <BatchNo>0059</BatchNo>
            <TerminalID>AMP00006</TerminalID>
        </ReportHeader>
        <ReportBody>
            <ReportLabel1>HOST TOTALS REPORT
            <CardTypeCount>6</CardTypeCount>
                <CardType_1>
                    <CardTypeLabel>MCRD</CardTypeLabel>
                    <RecordCount>1</RecordCount>
                    <Amount>$          1.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_1>
                <CardType_2>
                    <CardTypeLabel>VISA</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_2>
                <CardType_3
                    <CardTypeLabel>DISC</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_3>
                <CardType_4>
                    <CardTypeLabel>DEBT</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_4>
                <CardType_5>
                    <CardTypeLabel>JCB</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_5>
                <CardType_6>
                    <CardTypeLabel>AMEX</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_6>
                <Total>$          1.00  </Total>
                <BatchRespText>== BATCH TOTALS MATCH
==</BatchRespText>
```

```
</ReportLabel1>
<ReportLabel2>TERMINAL TOTALS REPORT
<CardTypeCount>2</CardTypeCount>
    <CardType_1>
        <CardTypeLabel1>MCRD</CardTypeLabel1>
        <Transaction_1>
            <TransType>Purchase/Compl</TransType>
            <RecordCount>1</RecordCount>
            <Amount>$          1.00</Amount>
        </Transaction_1>
        <Transaction_2>
            <TransType>Refund</TransType>
            <RecordCount>0</RecordCount>
            <Amount>$          0.00</Amount>
        </Transaction_2>
        <Transaction_3>
            <TransType>Void Purchase</TransType>
            <RecordCount>0</RecordCount>
            <Amount>$          0.00</Amount>
        </Transaction_3>
        <Transaction_4>
            <TransType>Void Refund</TransType>
            <RecordCount>0</RecordCount>
            <Amount>$          0.00</Amount>
        </Transaction_4>
        <Totals>
            <RecordCount>1</RecordCount>
            <Amount>$          1.00</Amount>
        </Totals>
    </CardType_1>
    <CardType_2>
        <CardTypeLabel2>VISA</CardTypeLabel2>
        <Transaction_1>
            <TransType>Purchase/Compl</TransType>
            <RecordCount>1</RecordCount>
            <Amount>$          1.00</Amount>
        </Transaction_1>
        <Transaction_2>
            <TransType>Refund</TransType>
            <RecordCount>0</RecordCount>
            <Amount>$          0.00</Amount>
        </Transaction_2>
        <Transaction_3>
            <TransType>Void Purchase</TransType>
            <RecordCount>0</RecordCount>
            <Amount>$          0.00</Amount>
        </Transaction_3>
        <Transaction_4>
            <TransType>Void Refund</TransType>
```

```
                        <RecordCount>0</RecordCount>
                        <Amount>$          0.00</Amount>
                    </Transaction_4>
                    <Totals>
                        <RecordCount>1</RecordCount>
                        <Amount>$          1.00</Amount>
                    </Totals>
                </CardType_2>
                <BatchTotals>
                    <BatchSaleCount>0</BatchSaleCount>
                    <BatchSaleAmount>$         0.00</BatchSaleAmount>
                    <BatchTipCount>0</BatchTipCount>
                    <BatchTipAmount>$          0.00</BatchTipAmount>
                    <BatchCashbackCount>0</BatchCashbackCount>
                    <BatchCashbackAmount>$
0.00</BatchCashbackAmount>
                    <BatchTaxCount>0</BatchTaxCount>
                    <BatchTaxAmount>$          0.00</BatchTaxAmount>
                    <BatchSurchargeCount>0</BatchSurchargeCount>
                    <BatchSurchargeAmount>$
0.00</BatchSurchargeAmount>
                    <BatchRefundCount>0</BatchRefundCount>
                    <BatchRefundAmount>$
0.00</BatchRefundAmount>
                    <BatchVoidSaleCount>0</BatchVoidSaleCount>
                    <BatchVoidSaleAmount>$
0.00</BatchVoidSaleAmount>
                    <BatchVoidRefundCount>0</BatchVoidRefundCount>
                    <BatchVoidRefundAmount>$
0.00</BatchVoidRefundAmount>
                    <BatchTotalAmount>$
0.00</BatchTotalAmount>
                </BatchTotals>
            </ReportLabel2>
            <ReportLabel3>SETTLEMENT REPORT
           <BatchRespText>SETTLEMENT SUCCESSFUL</BatchRespText>
            </ReportLabel3>
        </ReportBody>
        <ReportFooter>
            <ReceiptFooter1>Footer1</ReceiptFooter1>
            <ReceiptFooter2>Footer2</ReceiptFooter2>
            <ReceiptFooter3>Footer3</ReceiptFooter3>
            <ReceiptFooter4>Footer4</ReceiptFooter4>
            <ReceiptFooter5>Footer5</ReceiptFooter5>
            <ReceiptFooter6>Footer6</ReceiptFooter6>
        </ReportFooter>
    </XMLReceipt>
</XMLResponse>
```

10.3.9 FORCE CLOSE BATCH/ CLEAR BATCH

This method calls BaseApp’s Force Close Batch / Clear Batch transaction.

Prototype	`“cmdType”:”CloseBatch”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data** **Additional Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The XML format of KEY_RESPONSE_XMLDATA varies per processor.

- The following format applies to the TSYS and GlobalPay processors:

```
<XMLResponse>
    <XMLCmdType>ClearBatch</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:54:25</TransTime>
            <MerchantID>887000002726</MerchantID>
            <TerminalID>71134104</TerminalID>
      <FromDate>03/21/2020</FromDate>
      <FromTime>05:31 AM</FromTime>
      <ToDate>03/21/2020</ToDate>
      <ToTime>05:31 AM</ToTime>
      <BatchNo>238</BatchNo>
```

```
       </ReportHeader>
        <ReportBody>
            <ReportCount>2</ReportCount>
            <Report_1>
            <ReportLabel>BATCH DETAIL REPORT</ReportLabel>
            <Transaction_1>
                 <TransDate>03/21/2020</TransDate>
                 <TransTime>02:54:25</TransTime>
                 <TransName>SALE</TransName>
                 <InvoiceNo>000075</InvoiceNo>
                 <AccountNumber>************0057</AccountNumber>
                 <AuthNumber>552442</AuthNumber>
                 <ReferenceNumber>008610500940</ReferenceNumber>
                 <TransAmount>$10.00</TransAmount>
                 <TotalAmount>$10.00</TransAmount>
            </Transaction_1>
             <Transaction_2>
                 <TransDate>03/21/2020</TransDate>
                 <TransTime>02:59:25</TransTime>
                 <TransName>SALE</TransName>
                 <InvoiceNo>000075</InvoiceNo>
                 <AccountNumber>************0057</AccountNumber>
                 <AuthNumber>552442</AuthNumber>
                 <ReferenceNumber>008610500941</ReferenceNumber>
                 <TransAmount>$10.00</TransAmount>
                 <TotalAmount>$0.00</TotalAmount>
                 <NetworkID>000000</NetworkID>
                 <ReversedAmount>$10.00</ReversedAmount>
            </Transaction_2>
            <TerminalTotals>
                  <BatchTotalRecords>2</BatchTotalRecords>
                  <TotalVoidCount>0</TotalVoidCount>
                  <NetSales>$10.00</NetSales>
</TerminalTotals>
</Report_1>
<Report_2>
<ReportLabel>BATCH TOTAL REPORT</ReportLabel>
            <CardTypeCount>1</CardTypeCount>
                 <CardType_1>
            <CardTypeLabel>D</CardTypeLabel>
            <Transaction_1>
                 <TransType>Sale</TransType>
                 <RecordCount>1</RecordCount>
                 <Amount>$5.00</Amount>
            </Transaction_1>
            <Transaction_2>
                 <TransType>Return</TransType>
                 <RecordCount>0</RecordCount>
                 <Amount>$0.00</Amount>
            </Transaction_2>
           <Transaction_3>
                 <TransType>Void Sale</TransType>
                 <RecordCount>0</RecordCount>
                 <Amount>$0.00</Amount>
            </Transaction_3>
           <Transaction_4>
                 <TransType>Void Return</TransType>
                 <RecordCount>0</RecordCount>
                 <Amount>$0.00</Amount>
            </Transaction_4>
           <Totals>
                 <RecordCount>1</RecordCount>
                 <Amount>$5.00</Amount>
           </Totals>
           </CardType_1>
        <BatchTotals>
            <BatchSaleCount>1</BatchSaleCount>
            <BatchSaleAmount>$10.00</BatchSaleAmount>
            <BatchRefundCount>0</BatchRefundCount>
            <BatchRefundAmount>$0.00</BatchRefundAmount>
            <BatchVoidSaleCount>0</BatchVoidSaleCount>
            <BatchVoidSaleAmount>$0.00</BatchVoidSaleAmount>
            <BatchVoidRefundCount>0</BatchVoidRefundCount>
            <BatchVoidRefundAmount>$0.00</BatchVoidRefundAmount>
           <BatchTotalAmount>0</BatchTotalAmount>
        </BatchTotals>
        </Report_2>
       </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

- The following format applies to the PayFacto processor:

```
<XMLResponse>
    <XMLCmdType>Transaction</XMLCmdType>
    <XMLBody>
        <TransType>Force Close Batch</TransType>
    </XMLBody>
    <XMLReceipt>
        <ReportHeader>
            <ReceiptHeader1>NONE</ReceiptHeader1>
            <ReceiptHeader2>NONE</ReceiptHeader2>
            <ReceiptHeader3>NONE</ReceiptHeader3>
            <ReceiptHeader4>NONE</ReceiptHeader4>
            <ReceiptHeader5>NONE</ReceiptHeader5>
            <ReceiptHeader6>NONE</ReceiptHeader6>
            <TransDate>04/22/19</TransDate>
            <TransTime>06:43:37</TransTime>
            <BatchNo>0059</BatchNo>
            <TerminalID>AMP00006</TerminalID>
        </ReportHeader>
        <ReportBody>
            <ReportLabel1>HOST TOTALS REPORT
            <CardTypeCount>6</CardTypeCount>
                <CardType_1>
                    <CardTypeLabel>MCRD</CardTypeLabel>
                    <RecordCount>1</RecordCount>
                    <Amount>$          1.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_1>
                <CardType_2>
                    <CardTypeLabel>VISA</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_2>
                <CardType_3>
                    <CardTypeLabel>DISC</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_3>
                <CardType_4>
                    <CardTypeLabel>DEBT</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_4>
                <CardType_5>
                    <CardTypeLabel>JCB</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_5>
                <CardType_6>
                    <CardTypeLabel>AMEX</CardTypeLabel>
                    <RecordCount>0</RecordCount>
                    <Amount>$          0.00</Amount>
                    <MatchFlag>=</MatchFlag>
                </CardType_6>
                <Total>$          1.00  </Total>
                <BatchRespText>== BATCH TOTALS MATCH
==</BatchRespText>
            </ReportLabel1>
            <ReportLabel2>TERMINAL TOTALS REPORT
            <CardTypeCount>1</CardTypeCount>
```

```
                <CardType_1>
                    <CardTypeLabel>MCRD</CardTypeLabel>
                    <Transaction_1>
                        <TransType>Purchase/Compl</TransType>
                        <RecordCount>1</RecordCount>
                        <Amount>$          1.00</Amount>
                    </Transaction_1>
                    <Transaction_2>
                        <TransType>Refund</TransType>
                        <RecordCount>0</RecordCount>
                        <Amount>$          0.00</Amount>
                    </Transaction_2>
                    <Transaction_3>
                        <TransType>Void Purchase</TransType>
                        <RecordCount>0</RecordCount>
                        <Amount>$          0.00</Amount>
                    </Transaction_3>
                    <Transaction_4>
                        <TransType>Void Refund</TransType>
                        <RecordCount>0</RecordCount>
                        <Amount>$          0.00</Amount>
                    </Transaction_4>
                    <Totals>
                        <RecordCount>1</RecordCount>
                        <Amount>$          1.00</Amount>
                    </Totals>
                </CardType_1>
                <BatchTotals>
                    <BatchSaleCount>0</BatchSaleCount>
                    <BatchSaleAmount>$
0.00</BatchSaleAmount>
                    <BatchTipCount>0</BatchTipCount>
                    <BatchTipAmount>$          0.00</BatchTipAmount>
                    <BatchCashbackCount>0</BatchCashbackCount>
                    <BatchCashbackAmount>$
0.00</BatchCashbackAmount>
                    <BatchTaxCount>0</BatchTaxCount>
                    <BatchTaxAmount>$          0.00</BatchTaxAmount>
                    <BatchSurchargeCount>0</BatchSurchargeCount>
                    <BatchSurchargeAmount>$
0.00</BatchSurchargeAmount>
                    <BatchRefundCount>0</BatchRefundCount>
                    <BatchRefundAmount>$
0.00</BatchRefundAmount>
                    <BatchVoidSaleCount>0</BatchVoidSaleCount>
                    <BatchVoidSaleAmount>$
0.00</BatchVoidSaleAmount>
                    <BatchVoidRefundCount>0</BatchVoidRefundCount>
                    <BatchVoidRefundAmount>$
0.00</BatchVoidRefundAmount>
                    <BatchTotalAmount>$
0.00</BatchTotalAmount>
                </BatchTotals>
            </ReportLabel2>
            <ReportLabel3>FORCE CLOSE BATCH REPORT
            <BatchRespText>FORCE CLOSE BATCH
SUCCESSFUL</BatchRespText>
            </ReportLabel3>
        </ReportBody>
        <ReportFooter>
            <ReceiptFooter1>Footer1</ReceiptFooter1>
            <ReceiptFooter2>Footer2</ReceiptFooter2>
            <ReceiptFooter3>Footer3</ReceiptFooter3>
            <ReceiptFooter4>Footer4</ReceiptFooter4>
            <ReceiptFooter5>Footer5</ReceiptFooter5>
            <ReceiptFooter6>Footer6</ReceiptFooter6>
        </ReportFooter>
    </XMLReceipt>
</XMLResponse>
```

10.3.10 SET TMS ID

This method calls BaseApp’s TMS ID setting function, which causes the corresponding UI to be displayed for data entry.

Prototype	`“cmdType”:”SetTMSID”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

10.3.11 SET TMS HOST ADDRESS

This method calls BaseApp’s TMS Host Address function, which causes the corresponding UI to be displayed for data entry.

Prototype	`“cmdType”:”SetTMSHost”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

10.3.12 TMS DOWNLOAD

This method calls BaseApp’s TMS Download function.

Prototype	`“cmdType”:”TMSDownload”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

10.3.13 PARAMETER SETTINGS

This method calls BaseApp's Parameter Settings Menu function, which causes the corresponding UI to be displayed for data entry.

Prototype	`"cmdType":"ParameterSetting"`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.3.14 MEDIA SELECTION

This method calls BaseApp's Media Selection setting, which causes the corresponding UI to be displayed for data entry.

Prototype	`"cmdType":"SelectMedia"`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION

Application Type	Attended

10.3.15 UPLOAD AUDIT LOGS

This method calls BaseApp's Upload Audit Logs function.

Prototype	`"cmdType":"TMSUploadAuditLogs"`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

10.3.16 TERMINAL DETAILS REPORT

This method calls BaseApp's Terminal Detail Report print function.

Prototype	`"cmdType":"BatchTerminalDetail"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA

(refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format of KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>BatchTerminalDetail</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
           <ReportLabel>TERMINAL DETAIL REPORT</ReportLabel>
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:54:25</TransTime>
            <MerchantID>887000002726</BatchNo>
            <TerminalID>71134104</TerminalID>
           <FromDate>03/21/2020</FromDate>
           <FromTime>05:31 AM</FromTime>
           <ToDate>03/21/2020</ToDate>
           <ToTime>05:31 AM</ToTime>
           <BatchNo>238</BatchNo>
        </ReportHeader>
        <ReportBody>
            <Transaction_1>
                 <TransDate>03/21/2020</TransDate>
                 <TransTime>02:54:25</TransTime>
                 <TransName>SALE</TransName>
                 <InvoiceNo>000075</InvoiceNo>
                 <AccountNumber>************0057</AccountNumber>
                 <AuthNumber>552442</AuthNumber>
                 <TotalAmount>$10.00</TransAmount>
            </Transaction_1>
           <Transaction_2>
                 <TransDate>03/21/2020</TransDate>
                 <TransTime>02:59:25</TransTime>
                 <TransName>SALE</TransName>
                  <InvoiceNo>000075</InvoiceNo>
                 <AccountNumber>************0057</AccountNumber>
                 <AuthNumber>552442</AuthNumber>
                 <ReferenceNumber>008610500941</ReferenceNumber>
                 <TransAmount>$10.00</TransAmount>
                 <TotalAmount>$0.00</TransAmount>
                 <NetworkID>000000</NetworkID>
                 <ReversedAmount>$10.00</ReversedAmount>
            </Transaction_2>
<TerminalTotals>
    <BatchTotalRecords>1</BatchTotalRecords>
    <TotalVoidCount>0</TotalVoidCount>
    <NetSales>$5.00</NetSales>
</TerminalTotals>
</ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.3.17 TERMINAL TOTAL REPORT

This method calls BaseApp's Terminal Total Report print function.

Prototype	`“cmdType”:”BatchTerminalTotals”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format of KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>BatchTerminalTotals</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
<ReportLabel>TERMINAL TOTALS REPORT</ReportLabel>
```

```
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:54:25</TransTime>
            <MerchantID>887000002726</BatchNo>
            <TerminalID>71134104</TerminalID>
            <FromDate>03/21/2020</FromDate>
            <FromTime>05:31 AM</FromTime>
            <ToDate>03/21/2020</ToDate>
            <ToTime>05:31 AM</ToTime>
            <BatchNo>238</BatchNo>
        </ReportHeader>
        <ReportBody>
            <CardTypeCount>1</CardTypeCount>
<CardType_1>
            <CardTypeLabel>MASTERCARD</CardTypeLabel>
            <Transaction_1>
                  <TransType>Sale</TransName>
                  <RecordCount>1</RecordCount>
                  <Amount>$5.00</Amount>
             </Transaction_1>
<Transaction_2>
                  <TransType>Return</TransName>
                  <RecordCount>0</RecordCount>
                  <Amount>$0.00</Amount>
             </Transaction_2>
<Transaction_3>
                  <TransType>Void Sale</TransName>
                  <RecordCount>0</RecordCount>
                  <Amount>$0.00</Amount>
             </Transaction_3>
<Transaction_4>
                  <TransType>Void Return</TransName>
                  <RecordCount>0</RecordCount>
                  <Amount>$0.00</Amount>
             </Transaction_4>
            <Totals>
                  <RecordCount>1</RecordCount>
                  <Amount>$5.00</Amount>
</Totals>
            </CardType_1>
        <BatchTotals>
            <BatchSaleCount>1</BatchSaleCount>
            <BatchSaleAmount>$10.00</BatchSaleAmount>
```

```
                <BatchRefundCount>0</BatchRefundCount>
                <BatchRefundAmount>$0.00</BatchRefundAmount>
                <BatchVoidSaleCount>0</BatchVoidSaleCount>
                <BatchVoidSaleAmount>$0.00</BatchVoidSaleAmount>
                <BatchVoidRefundCount>0</BatchVoidRefundCount>
                <BatchVoidRefundAmount>$0.00</BatchVoidRefundAmount>
<BatchTotalAmount >0</BatchTotalAmount>
        </BatchTotals>
</ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.3.18 ACQUIRER DETAIL REPORT

This method calls BaseApp’s Acquirer Detail Report print function.

Prototype	`“cmdType”:”AcquirerDetail”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>AcquirerDetailReport</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
            <ReportLabel>ACQUIRER DETAIL REPORT</ReportLabel>
<ReceiptHeader1>AMP TEMP TERMINAL TESTING</ReceiptHeader1>
            <ReceiptHeader2>15    WERTHEIM    CRT.    UNITS    401-
403</ReceiptHeader2>
```

```
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP           TESTING           KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/2019</TransDate>
            <TransTime>02:58:15</TransTime>
        </ReportHeader>
        <ReportBody>
           <ReportLabel1>Acquirer Information</ReportLabel1>
            <AcquirerInfo>
                <AcquirerName>CTPAY</AcquirerName>
                <MerchantID>AMP0000001</MerchantID>
                <TerminalID>AMP00006</TerminalID>
            </AcquirerInfo>
           <ReportLabel2>Host Information</ReportLabel2>
            <Host_1>
                <Trx>
                    <IP_1>
                        <Address>TEST.CTPAIEMENT.CA</Address>
                        <Port>9885</Port>
                        <SSL>Y</SSL>
                        <SSLMethod>6</SSLMethod>
                        <SSLCert>ENTRUST_EV.CA</SSLCert>
                        <ClientAuth>N</ClientAuth>
                        <ClientKey>NONE</ClientKey>
                        <ClientCert>NONE</ClientCert>
                        <ClientPwd>NONE</ClientPwd>
                        <BaudRate>1200</BaudRate>
                        <DialMode>DTMF</DialMode>
                    </IP_1>
                    <IP_2>
                        <Address>TEST.CTPAIEMENT.CA</Address>
                        <Port>9885</Port>
                        <SSL>Y</SSL>
                        <SSLMethod>6</SSLMethod>
                        <SSLCert>ENTRUST_EV.CA</SSLCert>
                        <ClientAuth>N</ClientAuth>
                        <ClientKey>NONE</ClientKey>
                        <ClientCert>NONE</ClientCert>
                        <ClientPwd>NONE</ClientPwd>
                        <BaudRate>1200</BaudRate>
                        <DialMode>DTMF</DialMode>
                    </IP_2>
                </Trx>
                <Settle>
                    <IP_1>
                        <Address>TEST.CTPAIEMENT.CA</Address>
                        <Port>9885</Port>
                        <SSL>Y</SSL>
                        <SSLMethod>6</SSLMethod>
                        <SSLCert>ENTRUST_EV.CA</SSLCert>
                        <ClientAuth>N</ClientAuth>
                        <ClientKey>NONE</ClientKey>
                        <ClientCert>NONE</ClientCert>
                        <ClientPwd>NONE</ClientPwd>
                        <BaudRate>1200</BaudRate>
                        <DialMode>DTMF</DialMode>
                    </IP_1>
                    <IP_2>
                        <Address>TEST.CTPAIEMENT.CA</Address>
                        <Port>9885</Port>
                        <SSL>Y</SSL>
                        <SSLMethod>6</SSLMethod>
                        <SSLCert>ENTRUST_EV.CA</SSLCert>
                        <ClientAuth>N</ClientAuth>
                        <ClientKey>NONE</ClientKey>
                        <ClientCert>NONE</ClientCert>
                        <ClientPwd>NONE</ClientPwd>
                        <BaudRate>1200</BaudRate>
                        <DialMode>DTMF</DialMode>
                    </IP_2>
                </Settle>
            </Host_1>
        </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.3.19 LAST EMV TRANSACTION DETAILS

This method calls BaseApp’s Last EMV Transaction print function.

Prototype	`“cmdType”:”LastChipTransaction”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XML_DATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA

	KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>LastTransEmvDetails</XMLCmdType>
    <XMLReceipt>
 <ReportHeader>
      <ReportLabel>ACQUIRER DETAIL REPORT</ReportLabel>
      <ReceiptHeader1>AMP TEMP TERMINAL TESTING</ReceiptHeader1>
      <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-403</ReceiptHeader2>
      <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
      <ReceiptHeader4>AMP TESTING KRISTENE CONCHA</ReceiptHeader4>
      <TransDate>06/26/2019</TransDate>
<TransTime>02:58:15</TransTime>
<TerminalID>71134104</TerminalID>
  </ReportHeader>
        <ReportBody>
            <PAN>
                <TagName>5A PAN</TagName>
                <TagValue>************0119</TagValue>
            </PAN>
            <AIP>
                <TagName>82 AIP</TagName>
                <TagValue>A0000000031010</TagValue>
           </AIP>
            <DedicatedFilename>
                <TagName>84 Dedicated file name</TagName>
            </DedicatedFilename>
            <TransactionDate>
                <TagName>9A Transaction Date</TagName>
                <TagValue>180719</TagValue>
            </TransactionDate>
            <TransactionTime>
                <TagName>9F21 Transaction Time</TagName>
                <TagValue>180719</TagValue>
            </TransactionTime>
            <TransactionType>
                <TagName>9C Transaction Type</TagName>
                <TagValue>00</TagValue>
            </TransactionType>
            <PanSeqNum>
                <TagName>5F34 Pan Seq Num</TagName>
                <TagValue>01</TagValue>
            </PanSeqNum>
            <TranCurrencyCode>
                <TagName>5F2A Tran Currency Code</TagName>
                <TagValue>0124</TagValue>
            </TranCurrencyCode>
            <AuthorizedAmount>
                <TagName>9F02 Amount, authorized</TagName>
                <TagValue>($ 10.00)</TagValue>
            </AuthorizedAmount>
            <OtherAmount>
                <TagName>9F03 Amount, Other</TagName>
                <TagValue>000000000000</TagValue>
            </OtherAmount>
            <ICCAppVerNum>
                <TagName>9F08 ICC App Ver Num</TagName>
            </ICCAppVerNum>
            <TermAppVerNum>
                <TagName>9F09 Term App Ver Num</TagName>
                <TagValue>0001</TagValue>
            </TermAppVerNum>
            <TermCountryCode>
                <TagName>9F1A Term Country Code</TagName>
                <TagValue>0124</TagValue>
            </TermCountryCode>
            <TermCapabilities>
                <TagName>9F33 Terminal Capabilities</TagName>
                <TagValue>E0 B0 C8</TagValue>
            </TermCapabilities>
            <CVMResults>
                <TagName>9F34 CVM Results</TagName>
                <TagValue>5E0300</TagValue>
            </CVMResults>
            <TerminalType>
                <TagName>9F35 Terminal Type</TagName>
                <TagValue>22</TagValue>
            </TerminalType>
            <ATC>
                <TagName>9F36 ATC</TagName>
                <TagValue>00DC</TagValue>
            </ATC>
            <UnpredictableNum>
                <TagName>9F37 Unpredictable Num</TagName>
                <TagValue>2E9740B3</TagValue>
            </UnpredictableNum>
            <IACDefault>
                <TagName>9F0D IAC Default</TagName>
                <TagValue>B050808800</TagValue>
            </IACDefault>
            <IACDenial>
                <TagName>9F0E IAC Denial</TagName>
                <TagValue>0000000000</TagValue>
```

```
            </IACDenial>
            <IACOnline>
                <TagName>9F0F IAC Online</TagName>
                <TagValue>B050809800</TagValue>
            </IACOnline>
            <TACDefault>
                <TagName>TAC Default</TagName>
                <TagValue>B050809800</TagValue>
            </TACDefault>
            <TACDenial>
                <TagName>TAC Denial</TagName>
                <TagValue>B050809800</TagValue>
            </TACDenial>
            <TACOnline>
                <TagName>TAC Online</TagName>
                <TagValue>B050809800</TagValue>
            </TACOnline>
        </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.3.20 LAST TRANSACTION

This method calls BaseApp’s Last Transaction function to read the last transaction that Baseapp communicated to Host.

Prototype	`“cmdType”:”LastTransaction”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	
Application Type	Attended, Unattended

10.3.21 SHOW ADMIN MENU

This method calls BaseApp’s Admin Menu.

Prototype	`“cmdType”:”ShowAdminMenu”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.3.22 SHOW DEVICE SETTINGS MENU

This method calls BaseApp’s Device Settings Menu.

Prototype	`“cmdType”:”ShowDeviceSettingsMenu”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.3.23 SHOW REPRINT MENU

This method calls BaseApp’s Reprint Menu.

Prototype	`“cmdType”:”ShowReprintMenu”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.3.24 CARD AUTHENTICATION

This method calls BaseApp’s Card Authentication function.

Prototype	`“cmdType”:”CardAuthentication”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Card Number (PAN)** **Expiry Date** **Card Security Code/ Card Verification Value**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.3.25 OPEN PRE-AUTH AUTHORIZATION REPORT

This method calls BaseApp's Open Pre-Auth Report function.

Prototype	`“cmdType”:”OpenPreAuthReport”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>OpenPreAuth</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
<ReportLabel>OPEN PRE-AUTH REPORT</ReportLabel>
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:42:51</TransTime>
            <BatchNo>0102</BatchNo>
 <MerchantID>887000002726</MerchantID>
            <TerminalID>AMP00006</TerminalID>
 <FromDate>03/21/2020</FromDate>
 <FromTime>05:31 AM</FromTime>
 <ToDate>03/21/2020</ToDate>
 <ToTime>05:31 AM</ToTime>
```

```
        </ReportHeader>
        <ReportBody>
            <Transaction_1>
                <TransDate>2019/06/26</TransDate>
                <TransTime>02:42:08</TransTime>
                <InvoiceNo>0000011</InvoiceNo>
                <AccountNumber>************0119</AccountNumber>
                <AuthNumber>076237</AuthNumber>
                <ReferenceNumber>917706660800</ReferenceNumber>
                <TransAmount>$          5.00</TransAmount>
            </Transaction_1>
                 <TransCount>1</TransCount>
        </ReportBody>
        <ReportTotals>
            <TotalRecords>1</TotalRecords>
            <TotalAmount>$          5.00</TotalAmount>
        </ReportTotals>
    </XMLReceipt>
</XMLResponse>
```

10.3.26 EMV FALLBACK REPORT

This method calls BaseApp's EMV Fallback Report function.

Prototype	`"cmdType":"EMVFallbackReport"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>EMVFallbackReport</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
             <ReportLabel>EMV FALLBACK REPORT</ReportLabel>
            <ReceiptHeader1>AMP TEMP TERMINAL 
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE 
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:42:51</TransTime>
            <BatchNo>0102</BatchNo>
            <TerminalID>AMP00006</TerminalID>
        </ReportHeader>
        <ReportBody>
            <Threshold>0.0%</Threshold>
<SerialNumber>82000000078</SerialNumber>
<TotalTrans>59</TotalTrans>
<FallbackPercent>2%</FallbackPercent>
<FallbackCount>1</FallbackCount>
         </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.3.27 SET CONFIGURATION TAG VALUE

This method sets the value for a specific tag in BaseApp’s configuration parameters.

Prototype	`“cmdType”:”SetTagValue”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Configuration Tag** **Configuration Value**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION

Application Type	Attended, Unattended

10.3.28 GET CONFIGURATION TAG VALUE

This method retrieves the value of a specific tag in the internal configuration of BaseApp.

Prototype	`“cmdType”:”GetTagValue”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Configuration Tag**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_TAGVALUE KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

10.3.29 GET CARD DATA

This method retrieves the card data from the transaction.

Prototype	`“cmdType”:”GetCardData”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response	KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2

(refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

10.3.30 GET TOKEN DATA

This method retrieves the token data after performing an online transaction.

Prototype	“cmdType”:”GetToken”
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_TOKEN_DATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE

	KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_CODE KEY_RESPONSE_TRANSREF
Application Type	Unattended

10.3.31 EMV CONFIGURATION REPORT

This method calls BaseApp's EMV Configuration Report function.

Prototype	`"cmdType":"EMVConfigReport"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType>EMVConfigurationReport</XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
 <ReportLabel>EMV CONFIGURATION REPORT</ReportLabel>
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP           TESTING           KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:42:51</TransTime>
            <BatchNo>0102</BatchNo>
            <TerminalID>AMP00006</TerminalID>
        </ReportHeader>
        <ReportBody>
           <AppIDInfo>
 <AID>A0000000041001O</AID>
 <AIDLabel>A0000000041001O</AIDLabel>
             <Contact>
                 <TermType>25</TermType>
                 <TermCap>60D8C8</TermCap>
                 <AddlCap>6000F05001</AddlCap>
                 <TacDefault>DC 00 00 20 00</TacDefault>
                 <TacDenial>00 10 00 00 00</TacDenial>
                 <TacOnline>FC E0 9C F8 00</TacOnline>
                 <TerminalCountryCode>840</TerminalCountryCode>
                 <TerminalCurrencyCode>840</TerminalCurrencyCode>
                 <TerminalCurrencyExpiry>02</TerminalCurrencyExpiry>
                 <AppVerNumPri>0x0001</AppVerNumPri>
                 <AppVerNumSec>N/A</AppVerNumSec>
                 <FloorLimit>00000000</FloorLimit>
                 <RSThreshold>000000</RSThreshold>
                 <MaxRSPercentage>63</MaxRSPercentage>
                 <RSPercentage>63</RSPercentage>
                 <PartialSelect>Y</PartialSelect>
                 <AllowFallback>Y</AllowFallback>
                 <AllowPINBypass>N</AllowPINBypass>
                 <AcquirerID>0000000000000000</AcquirerID>
 </Contact>
<Contactless>
<TACDefault>F4 50 84 80 0C</TACDefault>
<TACDenial>00 00 00 00 00</TACDenial>
<TACOnline>F4 50 84 80 0C</TACOnline>
<TermCapCVMR>40</TermCapCVMR>
<TermCapCVMN>08</TermCapCVMN>
<FloorLimit>000000000000</FloorLimit>
<TransLimit>000000099999</TransLimit>
<ReqCVMLimit>00000050000<ReqCVMLimit>
<MChipEnable>Y</MChipEnable>
<MSDEnable>Y</MSDEnable>
</Contactless>
           <CAPKInfo>
           <CAPKList_1>
                 <Index>00</Index>
                 <ExpDate>311120</ExpDate>
</CAPKList_1>
```

```
</CAPKInfo>
</AppIDInfo>
         </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.3.32 GETTING TRANSACTION RESULTS

The third-party application should regularly call this function (right after the payment transaction’s API call) in order to get the transaction results.

Prototype	`“cmdType”:”GetTransactionResult”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	None
Application Type	Attended, Unattended

Example:

- Request:

```
http://192.168.0.108:22222/TRANSACTION
{"cmdType":" GetTransactionResult "}
```

- Response

```
{"responseAID":"","responseARC":"","responseAppPrefName":"","respons
eAuthCode":"","responseBaseAmount":"599","responseBatchNum":"1","res
ponseCBAmount":"","responseCVM":"","responseCardHolderName":"","resp
onseCardLabel":"","responseClerkNum":"NONE","responseCode":"","respo
nseCurrency":"$","responseCustomerServicePhone":"","responseDate":""
,"responseEntryMode":"","responseHostTimeStamp":"","responseIAD":"",
"responseInvoiceNum":"","responseMID":"AMP0000001","responseMerchant
Address1":"NONE","responseMerchantAddress2":"NONE","responseMerchant
Name":"NONE","responsePAN":"","responseResultCode":"-
1002","responseSCAmount":"","responseSequenceNumber":"","responseSto
reNum":"","responseTC":"","responseTID":"AMP00023","responseTSI":"",
"responseTVR":"","responseText":"","responseTime":"","responseTipAmo
unt":"","responseTotalAmount":"599","responseTransName":"PURCHASE","
responseTransRefNum":"","responseUserDefinedEchoData":"123456789","r
esponseXMLData":""}
```

In case there are no transaction results available at the time of the request, the payment app's return response should be:

Response:

```
{"AMPConnectResponseCode":4}
```

10.4HOST SPECIFIC TRANSACTIONS

The following list describes the processor specific methods that can be triggered from an external application.

Please refer to the following legend for the exceptions:

***Response values are only applicable to declined transactions in TSYS BaseApp.

10.4.1 TSYS PROCESSOR

1) **Terminal Initialization (Terminal Login).** This method calls the TSYS BaseApp's Terminal Initialization function.

Prototype	`“cmdType”:”TerminalInitialization”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

2) **Terminal Logout.** This method calls the TSYS BaseApp's Terminal Logout function.

Prototype	`"cmdType":"TerminalLogout"`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

3) **Re-enter Transaction.** This transaction is only available in a TSYS application that is configured to accept a re-enter transaction. The Purchase command that is described in section **10.3.1** is the same command that is used to perform the re-enter transaction.

Prototype	`"cmdType":"Purchase"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Card Entry Method*** **Expiry Date*** **Card Security Code/ Card Verification Value*** **Base Amount****
Optional Fields (See **Transaction Input Tags** for details)	**Tip Amount** **Address** **Zip Code** **Echo Data** **Cashback Amount** **Tax Amount**

Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN (unmasked) KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_CASHBKAMT KEY_RESPONSE_SURCHARGEAMT KEY_RESPONSE_TOTALAMT

	KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_TOKEN_DATA KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5

	KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_CASHBACKFEE KEY_RESPONSE_NONCASHFEE KEY_RESPONSE_EMV_SETTLEDATE KEY_RESPONSE_EMV_NETWORKID KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.4.2 PAYFACTO PROCESSOR

1) **Terminal Initialization.** This method calls the PayFacto BaseApp's Terminal Initialization (Key Exchange, Host Parameter Download, and EMV Download) function.

Prototype	`"cmdType":"TerminalInitialization"`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

2) **Key Exchange.** This method calls the PayFacto BaseApp's Key Exchange function.

Prototype	`"cmdType":"KeyExchange"`

Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

3) **Host Parameter Download.** This method calls the PayFacto BaseApp's Host Parameter Download function.

Prototype	`“cmdType”:”DLParam”`
Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

4) **EMV Download.** This method calls the Payfacto BaseApp's EMV Parameter Download function.

Prototype	`“cmdType”:”DLEMV”`

Mandatory Fields (See **Transaction Input Tags** for details)	None
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

5) **Host Detail Report.** This method calls the PayFacto BaseApp's Host Detail Report print function.

Prototype	`“cmdType”:”BatchHostDetail”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmdType> BatchHostDetail </XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>07/31/18</TransDate>
            <TransTime>00:58:44</TransTime>
            <BatchNo>0140</BatchNo>
            <TerminalID>AMP00001</TerminalID>
        </ReportHeader>
        <ReportBody>
            <CardType_1>
                <CardTypeLabel>VISA</CardTypeLabel>
                <Invoice_1>
                    <InvoiceNo>0000002</InvoiceNo>
                    <Amount>$10.00</Amount>
                </Invoice_1>
            </CardType_1>
        </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

6) **Host Total Report.** This method calls the PayFacto BaseApp's Host Totals Report print function.

Prototype	`“cmdType”:”BatchHostTotals”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response	KEY_RESPONSE_XMLDATA KEY_RESPONSE_RESULTCODE

(refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_APP_VERSION
Application Type	Attended, Unattended

The following describes the XML format in KEY_RESPONSE_XMLDATA:

```
<XMLResponse>
    <XMLCmd>MIS</XMLCmd>
    <XMLCmdType> BatchHostTotals </XMLCmdType>
    <XMLReceipt>
        <ReportHeader>
            <ReceiptHeader1>AMP TEMP TERMINAL
TESTING</ReceiptHeader1>
            <ReceiptHeader2>15 WERTHEIM CRT. UNITS 401-
403</ReceiptHeader2>
            <ReceiptHeader3>RICHMOND HILL ON L4B3H7</ReceiptHeader3>
            <ReceiptHeader4>AMP TESTING KRISTENE
CONCHA</ReceiptHeader4>
            <TransDate>06/26/19</TransDate>
            <TransTime>02:12:02</TransTime>
            <BatchNo>0101</BatchNo>
            <TerminalID>AMP00006</TerminalID>
            <ReportLabel>HOST TOTALS REPORT</ReportLabel>
        </ReportHeader>
        <ReportBody>
            <CardTypeCount>6</CardTypeCount>
            <CardType_1>
                <CardTypeLabel>MCRD</CardTypeLabel>
                <RecordCount>7</RecordCount>
                <Amount>$         30.00</Amount>
                <MatchFlag>=</MatchFlag>
            </CardType_1>
            <CardType_2>
                <CardTypeLabel>VISA</CardTypeLabel>
                <RecordCount>0</RecordCount>
                <Amount>$          0.00</Amount>
                <MatchFlag>=</MatchFlag>
            </CardType_2>
            <CardType_3>
                <CardTypeLabel>DISC</CardTypeLabel>
                <RecordCount>0</RecordCount>
                <Amount>$          0.00</Amount>
                <MatchFlag>=</MatchFlag>
            </CardType_3>
            <CardType_4>
                <CardTypeLabel>DEBT</CardTypeLabel>
                <RecordCount>0</RecordCount>
                <Amount>$          0.00</Amount>
                <MatchFlag>=</MatchFlag>
            </CardType_4>
            <CardType_5>
                <CardTypeLabel>JCB</CardTypeLabel>
                <RecordCount>0</RecordCount>
                <Amount>$          0.00</Amount>
                <MatchFlag>=</MatchFlag>
            </CardType_5>
            <CardType_6>
                <CardTypeLabel>AMEX</CardTypeLabel>
                <RecordCount>0</RecordCount>
                <Amount>$          0.00</Amount>
                <MatchFlag>=</MatchFlag>
            </CardType_6>
            <Total>$         30.00  </Total>
            <BatchRespText>== BATCH TOTALS MATCH ==</BatchRespText>
        </ReportBody>
    </XMLReceipt>
</XMLResponse>
```

10.4.3 GLOBALPAY PROCESSOR

1) **Transaction Adjust.** This method calls BaseApp's Transaction Adjust transaction.

Prototype	`"cmdType":"TransactionAdjust"`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Base Amount** **Adjust Amount Type** **Adjust Category Type** **Adjust Number**
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2

	KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2 KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_BASEAMT KEY_RESPONSE_TIPAMT KEY_RESPONSE_TOTALAMT KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM

	KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_APPPREFNAME KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5 KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_TRANSID KEY_RESPONSE_APP_VERSION KEY_RESPONSE_ISSUERNAME KEY_RESPONSE_GATEWAYTRANSID

	KEY_RESPONSE_BANKRESPONSECODE
Application Type	Attended

2) **Balance Inquiry.** This method calls BaseApp's Balance Inquiry transaction.

Prototype	`“cmdType”:”BalanceInquiry”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Card Entry Method*** **Expiry Date*** **Card Security Code/ Card Verification Value***
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response (refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_MERCH_NAME KEY_RESPONSE_MERCH_ADD1 KEY_RESPONSE_MERCH_ADD2 KEY_RESPONSE_CUST_SERV_PHONE KEY_RESPONSE_TIME KEY_RESPONSE_DATE KEY_RESPONSE_TID KEY_RESPONSE_MID KEY_RESPONSE_TRANSNAME KEY_RESPONSE_CARD_HOLDER_NAME KEY_RESPONSE_CARDLABEL KEY_RESPONSE_PAN KEY_RESPONSE_EXPIRYDATE KEY_RESPONSE_TRACK1 KEY_RESPONSE_TRACK2

	KEY_RESPONSE_ENTRYMODE KEY_RESPONSE_CVM KEY_RESPONSE_INVOICENUM KEY_RESPONSE_CLERKID KEY_RESPONSE_STORENUM KEY_RESPONSE_TRACENUM KEY_RESPONSE_BATCHNUM KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_TVR KEY_RESPONSE_AID KEY_RESPONSE_AUTHCODE KEY_RESPONSE_SEQUENCENUM KEY_RESPONSE_CURRENCY KEY_RESPONSE_IAD KEY_RESPONSE_TSI KEY_RESPONSE_ARC KEY_RESPONSE_CODE KEY_RESPONSE_TEXT KEY_RESPONSE_TC KEY_RESPONSE_HOSTTIMESTAMP KEY_RESPONSE_TRANSREF KEY_RESPONSE_RESULTCODE KEY_RESPONSE_HEADER1 KEY_RESPONSE_HEADER2 KEY_RESPONSE_HEADER3 KEY_RESPONSE_HEADER4 KEY_RESPONSE_HEADER5

	KEY_RESPONSE_HEADER6 KEY_RESPONSE_FOOTER1 KEY_RESPONSE_FOOTER2 KEY_RESPONSE_FOOTER3 KEY_RESPONSE_FOOTER4 KEY_RESPONSE_FOOTER5 KEY_RESPONSE_FOOTER6 KEY_RESPONSE_APP_VERSION KEY_RESPONSE_CODE_ISO KEY_RESPONSE_ISSUERNAME KEY_RESPONSE_GATEWAYTRANSID KEY_RESPONSE_BANKRESPONSECODE KEY_RESPONSE_AVAILABLE_AMT
Application Type	Attended

3) **Account Verification (Card Verify).** This method calls BaseApp's Card Verification transaction.

Prototype	`“cmdType”:”Purchase”`
Mandatory Fields (See **Transaction Input Tags** for details)	**Auto-Print** **Card Entry Method*** **Expiry Date*** **Card Security Code/ Card Verification Value***
Optional Fields (See **Transaction Input Tags** for details)	**Echo Data**
Key Response	KEY_RESPONSE_USER_DEFINED_ECHO_DATA KEY_RESPONSE_CODE

(refer to **Appendix A: Response Tags** for the description of each key)	KEY_RESPONSE_TEXT KEY_RESPONSE_RESULTCODE KEY_RESPONSE_APP_VERSION
Application Type	Attended

10.5PRINTING RESPONSE VALUES

To print the details of a transaction completed by BaseApp, **Auto-Print** must be set to true. Refer to **Transaction Input Tags** for more details.

If the value is set to false, no printing will be performed in BaseApp. This means printing is handled by the client application, if it is needed.

For the client application to handle printing, the Android project must be configured properly as described in the “AMP POS 6 and 8 Series Developer Guide” document, “Section 3 - Adding the AMP POS 6 and 8 Series libraries to the Project”. Without the mentioned libraries, the device printing functions will not be available.

Please refer to the AMP Android API documentation, provided in the AMP 8 Series SDK package, for the complete list of available printer API. The printing of the response values is exclusive to attended BaseApp.

11 SYSTEM

This section addresses the System API calls and provides information about the commands that get terminal information.

The following URL should be used for all the API calls for system command.

Endpoint URL: SYSTEM.

Example: http://IP_Address:22222/SYSTEM

11.1 SYSTEM COMMANDS

11.1.1 GETTING TERMINAL INFO

Getting Terminal Info			
JSON key	**JSON key description**	**Value type**	**Value**

cmdType	Command type	String	"GET_TERMINAL_INFO"

The response is a String text which is in a String array format. The content is in the following order:

"App name", "App version", "Build Type", "Firmware version", "Terminal Serial Number" and "Terminal Date and time".

11.1.2 REBOOTING THE TERMINAL

Rebooting the Terminal			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type	String	" REBOOT_TERMINAL"

Note: The AMP Connect app executes device reboot with a 5 second delay after receiving the reboot request command. This is to ensure that the response command will be sent back to the client app.

12 TRANSACTION INPUT TAGS

Some transactions require field setting prior to performing a transaction. The following are the available methods for setting the input fields values.

12.1.1 AUTO-PRINT

Prototype	`"AutoPrint":""`
Description	The method is used to set the external application printing status
JSON Key	AutoPrint
JSON Value	true – BaseApp will print receipt false – BaseApp will not print receipt and the business app should
Notes	This field is set to FALSE automatically for unattended applications

12.1.2 CARD ENTRY METHOD

Prototype	`“CardEntryMethod”:””`
Description	The method is used to set the card entry method that will be used in the transaction
JSON Key	CardEntryMethod
JSON Value	AUTO – the card is expected to be swiped, tapped, or inserted in the card slot MANUAL – the card details are expected to be entered manually
Notes	This field is automatically set to AUTO for unattended applications

12.1.3 CARD NUMBER (PAN)

Prototype	`“CardEntryMethodCardNumber”:””`
Description	The method is used to set the Primary Account Number (PAN) that will be used in the transaction
JSON Key	CardEntryMethodCardNumber
JSON Value	String number – card PAN represented in string format
Notes	The setting is required when the card entry method is set to MANUAL; Not required when the card entry method is set to AUTO

12.1.4 EXPIRY DATE

Prototype	`“EntryMethodExpiryDate”:””`
Description	The method is used to set the expiry date that will be used in the transaction
JSON Key	EntryMethodExpiryDate
JSON Value	String date – indicates the expiration date of the card in MM/YY string format
Notes	The setting is required when the card entry method is set to MANUAL; Not required when the card entry method is set to AUTO

12.1.5 CARD SECURITY CODE/ CARD VERIFICATION VALUE

Prototype	`“CardEntryMethodVcode”:””`
Description	The method is used to set the code for “card not present” payment card transactions. The term for this code varies across different card brands: • "CSC" or "Card Security Code" – American Express • "CVC2" or "Card Validation Code" – MasterCard • "CVD" or "Card Verification Data" – Discover • "CVN2" or "Card Validation Number 2" – China UnionPay • "CVV2" or "Card Verification Value 2" – Visa • "CVC" or "Card Verification Code" – MasterCard
JSON Key	CardEntryMethodVcode
JSON Value	String code – indicates the 3 or 4-digit code of the card to be used in the transaction. The code length depends on the card brand: AMEX – accepts 3 or 4 digits The rest of the card brands (Visa, MasterCard, etc.) – 3 digits
Notes	When the card entry method is set to MANUAL, the setting is optional, depending on the Acquirer. Not required when the card entry method is set to AUTO.

12.1.6 ADDRESS

Prototype	`“CardEntryMethodAddress”:””`
Description	The method is used to set the card holder address.
JSON Key	CardEntryMethodAddress
JSON Value	String address – indicates the card holder address. This value can be set up to 256 characters.
Notes	The setting is optional when the card entry method is set to MANUAL; Not required when the card entry method is set to AUTO.

12.1.7 ZIP CODE

Prototype	`“CardEntryMethodZipCode”:””`

Description	The method is used to set the card holder’s address zip code.
JSON Key	CardEntryMethodZipCode
JSON Value	String zip – indicates the card holder’s address zip code. This value can be set up to 256 characters.
Notes	The setting is optional when the card entry method is set to MANUAL; Not required when the card entry method is set to AUTO.

12.1.8 ECHO DATA

Prototype	`UserDefinedEchoData`
Description	The method is used to set a unique identifier to be used by the external application for its internal processes.
JSON Key	UserDefinedEchoData
JSON Value	String brn – a 50-byte character that represents a unique identifier for the transaction.
Notes	The setting is optional for any financial transaction (e.g. Sale, Refund, Force Post, etc.)

12.1.9 BASE AMOUNT

Prototype	`BaseAmount`
Description	The method is used to set the transaction base amount.
JSON Key	BaseAmount
JSON Value	String amt – an 8-digit numeric value in string format that represents the transaction base amount. The general format of this value is “999999.99”. If presented differently, BaseApp modifies the value. For example: • 1.23 is reflected as 1.23 • 123 is reflected as123.00 • 12.3 is reflected as 12.30 • 123. is reflected as 123.00 • 1.234 is reflected as 1.23 • 1.235 is reflected as 1.24

Notes	The setting is required for some financial transactions, e.g. Sale, Refund, Pre-Authorization.

12.1.10 TIP AMOUNT

Prototype	`TipAmount`
Description	The method is used to set the transaction tip amount.
JSON Key	TipAmount
JSON Value	String amt – an 8-digit numeric value in string format that represents the transaction tip amount. The general format of this value is "999999.99". If presented differently, BaseApp modifies the value. For example: • 1.23 is reflected as 1.23 • 123 is reflected as123.00 • 12.3 is reflected as 12.30 • 123. is reflected as 123.00 • 1.234 is reflected as 1.23 • 1.235 is reflected as 1.24
Notes	The setting is optional for some financial transactions, e.g. Sale, Refund, Pre-Authorization.

12.1.11 CASHBACK AMOUNT

Prototype	`CashbackAmount`
Description	Method is used to set the transaction cashback amount.
JSON Key	CashbackAmount
JSON Value	String cashback – an 8-digit numeric value in string format that represents the transaction cashback amount. The general format of this value is "999999.99". If presented differently, BaseApp modifies the value. For example: • 1.23 is reflected as 1.23 • 123 is reflected as123.00 • 12.3 is reflected as 12.30 • 123. is reflected as 123.00 • 1.234 is reflected as 1.23 • 1.235 is reflected as 1.24

Notes	The setting is optional for some financial transactions, e.g. Sale.

12.1.12 TAX AMOUNT

Prototype	`TaxAmount`
Description	The method is used to set the transaction tax amount.
JSON Key	TaxAmount
JSON Value	String tax – an 8-digit numeric value in string format that represents the transaction tax amount. The general format of this value is “999999.99”. If presented differently, BaseApp modifies the value. For example: • 1.23 is reflected as 1.23 • 123 is reflected as123.00 • 12.3 is reflected as 12.30 • 123. is reflected as 123.00 • 1.234 is reflected as 1.23 • 1.235 is reflected as 1.24
Notes	The setting is optional for Sale transactions.

12.1.13 AUTHORIZATION CODE

Prototype	`AuthCode`
Description	The method is used to set the transaction authorization code.
JSON Key	AuthCode
JSON Value	String amt – a 6-digit alphanumeric string that represents the transaction authorization code.
Notes	The setting is required for Force Post transactions.

12.1.14 TRANSACTION ID

Prototype	`TransID`
Description	The method is used to set the Transaction ID.
JSON Key	TransID

JSON Value	String amt – a 15-digit alphanumeric string that represents the transaction ID.
Notes	The setting that is required for Force Post transactions, depending on the processor.

12.1.15 VOID NUMBER

Prototype	`VoidNumber`
Description	The method is used to set the transaction's void number.
JSON Key	VoidNumber
JSON Value	String in – a numeric value in string format that represents the void number. Void types that require an amount as input are expected to follow the "999999.99" format. If presented differently, BaseApp modifies the value. For example: • 1.23 is reflected as 1.23 • 123 is reflected as123.00 • 12.3 is reflected as 12.30 • 123. is reflected as 123.00 • 1.234 is reflected as 1.23 For the void types that require numbers as input, the values are validated according to the Acquirer requirements.
Notes	The setting is required for Void transactions.

12.1.16 VOID TYPE

Prototype	`VoidType`
Description	The method is used to set the transaction's void type.
JSON Key	VoidType
JSON Value	VOID_TYPE vt – type of void transaction to be performed: • INVOICE_NUMBER • TRACE_NUMBER • RRN_NUMBER • AUTH_NUMBER • CARD_NUMBER • FULL_BY_AMT

	- FULL_BY_RRN - PARTIAL_BY_AMT - PARTIAL_BY_RRN
Notes	The setting is required for Void transactions.

12.1.17 COMPLETION NUMBER

Prototype	`CompletionNumber`
Description	The method is used to set the transaction's completion number.
JSON Key	CompletionNumber
JSON Value	String number – a numeric value in string format that represents the completion number. The values are validated according to the Acquirer requirements.
Notes	The setting is required for Completion transactions, except when the completion type is LAST_TRANSACTION.

12.1.18 COMPLETION TYPE

Prototype	`CompletionType`
Description	The method is used to set the transaction completion type.
JSON Key	CompletionType
JSON Value	COMPLETION_TYPE cp – type of completion transaction to be performed: - LAST_TRANSACTION - TRACE_NUMBER / SEQUENCE_NUMBER - INVOICE_NUMBER - REFERENCE_NUMBER - AUTH_NUMBER - GATEWAYID
Notes	The setting is required for Completion transactions.

12.1.19 STANDALONE MODE VALUE

Prototype	`StandaloneMode`

Description	The method is used to set the application's standalone mode status.
JSON Key	StandaloneMode
JSON Value	true – BaseApp can run as a standalone application. false – BaseApp cannot run unless triggered by the external application.
Notes	The setting is required for standalone setting command. Standalone Mode value for unattended applications is fixed to FALSE.

12.1.20 CONFIGURATION TAG

Prototype	`Tag`
Description	The method is used to set the tag to be configured in BaseApp configuration file.
JSON Key	Tag
JSON Value	Indicates the value that will be set in the configuration file
Notes	

12.1.21 CONFIGURATION VALUE

Prototype	`Value`
Description	The method is used to set the value of the tag to be configured in the BaseApp configuration file.
JSON Key	Value
JSON Value	Indicates the value that will be set in the configuration file.
Notes	

12.1.22 ADJUST AMOUNT TYPE

Prototype	`AdjustAmountType`
Description	The method is used to set the type of amount to be adjusted using the Transaction Adjust command.

JSON Key	AdjustAmountType
JSON Value	The amount type to be adjusted: • PREAUTH – Pre-Authorized amount will be adjusted • TIP – Tip will be adjusted • TAX – Tax amount will be adjusted
Notes	This setting is only applicable to GlobalPay's Transaction Adjust command.

12.1.23 ADJUST CATEGORY TYPE

Prototype	`AdjustCategoryType`
Description	The method is used to set the Transaction Adjust retrieval option.
JSON Key	AdjustCategoryType
JSON Value	Transaction Adjust retrieval option: • LAST_TRANSACTION – last transaction will be adjusted • INVOICE_NUMBER – transaction with the specified invoice number will be adjusted • AUTH_NUMBER – transaction with the specified Authorization Code will be adjusted • CARD_NUMBER – transaction with the specified card number will be adjusted • GATEWAYID – transaction with the specified Gateway Transaction ID will be adjusted
Notes	This setting is only applicable to the GlobalPay's Transaction Adjust command.

12.1.24 ADJUST NUMBER

Prototype	`AdjustNumber`
Description	The method is used to set the transaction adjust numeric value.
JSON Key	AdjustNumber
JSON Value	A numeric value in string format that represents the adjust number. The value depends on the "Adjust Category Type" supplied (see **Adjust Category Type** for the list of possible adjust category types).

Notes	This setting is only applicable to GlobalPay's Transaction Adjust command.

12.1.25 ADDITIONAL DATA

Prototype	`AdditionalData`
Description	The value is used to send any other data that is required by the payment application's transaction, such as Fleet card or bill payment extra data elements.
JSON Key	AdditionalData
JSON Value	A JSON string.
Notes	Refer to format of JSON for this filed based on the Host and transaction options.

13 RESPONSE CODES

13.1 AMP CONNECT RESPONSE CODE

The following table includes the communication response codes that are returned from the AMP Connect app.

AMP Connect Response Codes	
"AMPConnectResponseCode" Tag	**Value**
0	Success
1	Invalid parameter
2	Invalid Command type
3	Invalid URI
4	No transaction result available
5	Server busy
6	Unsupported http method
7	A device's incompatible feature
8	BaseApp has not been installed on the terminal

9	Payment option was not selected
10	3rd party payment app is not available
21	Invalid display data; exclusive to Display end point
22	QR code conversion failure; exclusive to Display end point

13.2 CLOUD RESPONSE CODE

The following table includes the communication response codes that are returned from the AMP Cloud.

AMPConnect server “canLoad” Tag	
Value	**Description**
true / false	Indicates success or failure of the operation

AMPConnect server “status” Tag	
Value	**Description**
INITIALIZED	Command is received from business application
SENT_PAYLOAD	Command is received by terminal
RECEIVED_RESPONSE	Response is sent from terminal
DONE	Response is fetched by business application

AMPConnect server “message” Tag	
Value	**Description**
string	A message returned by the server to describe the command execution status. The business app can display this message to the user.

13.3 BASEAPP RESPONSE RESULT CODES

The following table defines the value of the parameter “responseResultCode” that is returned by BaseApp in “Payload” tag. When this tag is present in the response, it means that the terminal received the transaction or command and it was processed. A “responseResultCode” value of “0000” indicates the transaction execution. The actual transaction execution result is determined by the tags, which are explained in **Appendix A: Response Tags**.

Response Result Codes	
“ResponseResultCode” Tag	**Value**
0000	Returned when transaction is successful.
-1001	Returned when cancel is pressed during transaction.
-1002	Returned when timeout occurs during transaction.
-1003	Generic Transaction error (invalid input).
-1005	Returned when record is not found
-1006	Returned when transaction is not allowed
-2001	Key exchange is not successful.
-2002	Parameter download is not successful.
-2003	EMV download is not successful.

14 AMP CONNECT AGENT

AMP Connect Agent is another AMP offering that facilitates integration of a business application with AMP payment application, BaseApp. AMP Connect Agent and the business application should run on the same device or terminal. The business app can run on a browser or run on an Android device.

The following features are provided by AMP Connect Agent:

- Supports API call directly from any browser.
- Supports all AMP Connect functions.
- Runs on an Android device and manages the transaction process through AMP Connect.
- Communicates with business app through REST API.
- Runs as a service and does not need to be launched upon consecutive device restarts.
- Functions are executed in a synchronous manner; therefore, the results are available after the function call.
- Prints text or image on terminal printer, a USB printer that is connected to the device, or a printer that is connected to IP/WIFI and supports ESCPOS commands.

14.1 MESSAGE FORMAT

The message format is described in the **Transaction Endpoint** and **Print endpoint**. All commands should be sent as POST.

14.2 GENERATING THE REST API URL ADDRESS

The IP address is always 127.0.0.1 (localhost) to create the REST API URL address. The port number is always fixed and is set to “33333”. The below pattern should be followed for each REST API call:

```
POST>“http://127.0.0.1:33333/” + “endpoint”
```

For instance, the URL address for a payment related API call should be:

```
http://127.0.0.1:33333/TRANSACTION
```

14.3 PRINT ENDPOINT

This end point is used to print text or image on different printers.

14.3.1 REQUESTS

The printer request is in JSON format. There are two mandatory JSON keys “cmdType” and “ReqPayload”. The parameters for each transaction type are defined inside the “ReqPayload” JSON object.

Transaction Request			
Request Format	`{“cmdType”:”, “ReqPayload”:{“”:””, “”:””,“”:””, ……}}`		
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: calling a payment transaction.	String	See **Print commands** for details.
ReqPayload	The request payload, which defines all transaction parameters	String (JSON Object)	See the following table for details.

The ReqPayload contains three tags that are described below, along with their embedded tags:

JSON key	JSON key	Value type	Value
PrinterData		String	Print data; Refer to **Print commands.**
PrintParm	PaperSize	String	“Small” for 2 inches “Large” for 3 inches
Connection	Connection Type	String	“Wlan” for IP printer “Usb” for USB printer “Bluetooth” for Bluetooth printer “Terminal” for AMP8000 and AMP8200 printers
	BluetoothName		Bluetooth name of the Bluetooth printer
	Ip		IP address of network printer
	Port		Port address of network printer
	VendorId		Vendor ID of USB printer
	ProductId		Product ID of USB printer

14.3.2 RESPONSE

The response to a request is in JSON and has the following tag:

JSON key	JSON key description	Value
responseCode	Please refer to **AMP Connect Agent Response Code**.	The response code of the print endpoint.

14.3.3 PRINT COMMANDS

1) Print Image

This method prints image on the printer.

Prototype	`“cmdType”:”PrintImage”`
Mandatory Fields	PrintData PrintParam ConnectType
Notes	The PrintData is in BASE64 format of the image that should be printed.

- Example:

```
{
    "cmdType": "PrintImage",
    "ReqPayload": {
        "PrintData": "xx",
        "PrintParam": {
            "PaperSize": "Small"
        },
        "Connection": {
            "ConnectType": "Bluetooth",
            "BluetoothName": "xx"
        }
    }
}
```

2) Print Text

This method prints text on the printer.

Prototype	`“cmdType”:”PrintText”`
Mandatory Fields	PrintData PrintParam ConnectType
Notes	The PrintData is the string to be printed. ESCPOS commands that are supported by the printer can be inserted into this string. A list of the most important ESCPOS commands are added to **Appendix F** for reference. Note: ESCPOS commands may vary between printers.

- Example IP printer:

```
{
    "cmdType": "PrintText",
    "ReqPayload": {
        "PrintData": "xx",
        "PrintParam": {
            "PaperSize": "Large"
        },
        "Connection": {
            "ConnectType": "Wlan",
            "Ip": "xx",
            "Port": "9100"
        }
    }
}
```

- Example USB printer:

```
{
    "cmdType": " PrintText",
    "ReqPayload": {
        "PrintData": " xx ",
        "PrintParam": {
            "PaperSize": "Small"
        },
        "Connection": {
            "ConnectType": "Usb",
            "VendorId": "1046",
            "ProductId": "20495"
```

```
                    }
                }
            }
```

14.4 TRANSACTION ENDPOINT

This endpoint sends a transaction to AMP payment application, BaseApp. Refer to **Transaction Commands** section to see the list of all commands.

The following URL should be used for all AMP Connect transaction API calls:

Endpoint URL: TRANSACTION.

http://127.0.0.1:33333/TRANSACTION

The following sections describe the common transaction API calls that can be triggered from another device for communicating with the AMP payment application.

14.4.1 TRANSACTION REQUESTS

The AMP Connect Agent data request is in JSON format. There are three mandatory JSON keys “TerminalId”, “cmdType” and “ReqPayload”. The parameters for each transaction type are defined inside the “ReqPayload” JSON object.

Transaction Request			
Request Format	`{“TerminalId”: ””,”cmdType”:””, “ReqPayload”:{“”:””, “”:””,“”:””, ……}}`		
JSON key	**JSON key description**	**Value type**	**Value**
TerminalId	The serial number of the terminal that executes the transaction.	String	
cmdType	The command type or the intended action: calling a AMP Connect transaction.	String	See **Transaction Commands** for details.
ReqPayload	The request payload, which defines all transaction parameters.	String (JSON Object)	See **Transaction Input Tags** for details.

- Sample request message:

```
{
    "TerminalId": "6700000109",
    "cmdType": "Purchase",
    "ReqPayload": {
        "BaseAmount": "14.2",
        "UserDefinedEchoData": "testdata",
        "CardEntryMethod": "AUTO",
        "AutoPrint": "false"
    }
}
```

14.4.2 TRANSACTION RESULT

The response data format is unified across all transaction commands. The following table includes the details.

JSON key	JSON key description	Value
EndPoint	The endpoint that is called.	Always is “TRANSACTION”
CmdType	This value should be the same as the ‘cmdType” in the request.	See **Transaction Commands** for details.
Status	The payment transaction’s status.	“RESULT_OK” “RESULT_CANCELLED”
UserDefinedEchoData	This value matches the data in the request.	
Payload	The actual transaction response data that is returned by the payment app.	

- Sample response message:

```
{"EndPoint":"TRANSACTION","CmdType":"Purchase","UserDefinedEchoData"
:"testdata","TransactionStatus":"RESULT_OK","Payload":{"response_tid
_key":"*****001","response_currency_key":"$","response_date_key":"08
172020","response_token_data":"","response_emv_transdate_key":"20081
7","response_emv_cid_key":"80","response_emv_networkid_key":"8926389
90850305","response_resultcode_key":"0000","response_merch_add1_key"
:"Merchant Address 1","response_merch_add2_key":"Merchant Address 2"
,"response_tsi_key":"0000","response_aid_key":"A0000000031010","resp
onse_emv_currencycode_key":"0124","response_tvr_key":"0000000000","r
esponse_emv_iac_online_key":"","response_invoicenum_key":"000001","r
esponse_emv_iac_denial_key":"","response_time_key":"185154","respons
e_pan_key":"452085******8131","response_emv_cvm_key":"3F0000","respo
nse_arc_key":"3035","response_cashbkamt_key":"","response_storenum_k
ey":"","response_baseamt_key":"1420","response_emv_iad_key":"","resp
onse_user_defined_echo_data":"testdata","response_code_iso_key":"","
response_cashbackfee_key":"","response_track1_key":null,"response_tr
ack2_key":"452085******8131=****201*************","response_totalamt
_key":"1420","response_bankresponsecode":"","response_noncashfee_key
":"","response_appprefname_key":null,"response_issuername":"VISA","r
esponse_text_key":"DECLINE","response_code_key":"05","response_seque
ncenum_key":null,"response_emv_settledate_key":"","response_card_hol
der_name_key":" \/","response_clerkid_key":"","response_expirydate_k
ey":null,"response_emv_hosttimestamp_key":"08\/17\/2020 21:51:59","r
esponse_authcode_key":"","response_emv_transtype_key":"00","response
_emv_usage_ctrl_key":"","response_footer1_key":"","response_footer2_
key":"","response_footer3_key":"","response_footer4_key":"","respons
e_footer5_key":"","response_footer6_key":"","response_header1_key":"
Merchant Name","response_header2_key":"Merchant Address 1","response
_header3_key":"Merchant Address 2","response_header4_key":"Merchant
Address 3","response_header5_key":"Customer Service Phone Number","r
esponse_header6_key":"123-456-
7890","response_entrymode_key":"P","response_emv_applabel_key":"VISA
 CREDIT","response_emv_app_pan_key":"452085******8131","response_emv
_tac_default_key":null,"response_batch_key":"","response_mid_key":"*
********001","response_cvm_key":"","response_channel_id_key":"","res
ponse_emv_aip_key":"0000","response_cardlabel_key":"CREDIT","respons
e_emv_tvr_key":"0000000000","response_emv_termcode_key":"0840","resp
onse_emv_iac_default_key":"","response_iad_key":"","response_tc_key"
:null,"response_available_amt_key":"","response_emv_ac_key":"A463DC2
FFCC99670","response_emv_atc_key":"01C6","response_emv_unpred_num_ke
y":"E427C07F","response_reversal_amt_key":"","response_cust_serv_pho
ne_key":"","response_transref_key":"","response_tipamt_key":"","resp
onse_transid_key":"","response_tagvalue_key":"","response_hosttimest
amp_key":"08\/17\/2020 21:51:59","response_merch_name_key":"Merchant
 Name","response_emv_tac_online_key":null,"response_app_version_key"
:"AMP POS v02.02.016.GPPhu01.00.001","response_emv_tac_denial_key":n
ull,"response_surchargeamt_key":"","response_transname_key":"Sale","
response_emv_oth_amt_key":"000000000000","response_gatewaytransid":"
1332315573","response_emv_app_transdata":null,"response_trace_key":"
000001","response_emv_auth_amt_key":"000000001420","response_emv_pan
seqnum_key":"01","response_xmldata_key":""}}
```

14.5 CUSTOMER DISPLAY ENDPOINT

This endpoint utilizes the customer display on AMP6700 through the AMP Connect Agent.

The following URL should be used for customer display API calls:

Endpoint: CUSTOMER_DISPLAY

http://127.0.0.1:33333/CUSTOMER_DISPLAY

14.5.1 REQUESTS

The customer display request is in JSON format. This request has the same format as AMP Connect's Customer Display, plus the additional "TerminalId" tag.

Transaction Request			
JSON key	**JSON key description**	**Value type**	**Value**
TerminalId	The serial number of the terminal that is used for customer display.	String	
OtherTags			Refer to the **Customer Display Commands** section

- Sample request message:

```
{
    "TerminalId": "6700000109",
    "amount": 3.99,
    "description": "Coca cola",
    "item_number": 123,
    "item_type": "NORMAL",
    "quantity": 3,
    "subtotal": 9.99,
    "tax": 1.00,
    "total": 10.99,
    "cmdType": "ADD_ITEM"
}
```

14.5.2 RESPONSE

The response to any request is in JSON and has the following tags.

JSON key	JSON key description	Value
responsCode	Please refer to **AMP Connect Agent Response Code**.	The response code of the customer display endpoint.

- Sample response message:

```
{"responseCode":0}
```

14.6 AMP CONNECT AGENT RESPONSE CODES

The following table includes the response codes that are returned from the AMP Connect Agent.

Agent Response Codes	
"responseCode" Tag	**Value**
0	Success
1	Invalid URL
2	Invalid parameter
3	Agent is busy and executing another task
4	Printer is not ready
5	Unsupported Http method
6	BaseApp is either not installed on the terminal or is not configured properly to accept a request.
7	The payment application launch failed.
8	Not supported. This function is not implemented.
9	Cannot connect to AMP Connect with the provided Terminal ID.
10	Print data error.
21	Invalid display data; exclusive to Display end point

15 DIGITAL RECEIPT

AMP Payment application provides a solution to generate identical receipts in the back office by providing a JSON receipt data which contains the appropriate formatting.

Note: Currently, this feature is supported in some AMP Payment Applications. Please contact your account manager to make sure the feature is supported in the application that will be used in your integration project.

15.1 JSON DATA

The JSON data for digital receipt are in “response_jsonrcpt_key” tag in transaction message response.

The returned JSON format contains the “layout” and “lines” details needed to generate a receipt.

15.1.1 LAYOUT

For “layout”, the mandatory fields that should be returned are the following:

- “format” sets the type of receipt format to be generated.
    - “condensed” – The format that prints receipts in an economical manner by saving space and paper.
    - “normal” – The format that prints receipts in a standard and easier to read format.
- “fontmaxchars” sets the font and maximum printed characters in a receipt.
    - “regular” – Sets font to a regular size; a maximum of 34 characters can be printed per line.
    - “emphasis” – Sets font to bold and larger than standard size; a maximum of 24 characters can be printed per line.
    - “detail” – Sets font to smaller than standard size; a maximum of 42 characters can be printed per line.

15.1.2 LINES

“lines” contains the type, alignment, and image/text values for each receipt detail per line.

- “type” – Value that identifies the type of detail printed per line.
    - “image” – Content of line is receipt logo.
    - “line” – Content of line is text details.
    - “nextline” – Content of line will be printed in the next line.
- “font” – Value for the type of font used per line.
    - “regular” – Content of line is in regular font.
    - “bold” – Content of line is in bold font.

  Note: This is an optional field.
- “align” – Value for the text alignment used per line.
    - “c” – Content of line is printed in center.
    - “l” – Content of line is printed left justified.
    - “r” – Content of line is printed right justified.
    - “lr” – Content of line will contain 2 text input and is printed left and right justified.

- “image” – Filename that is used for receipt logo.
- “text” – Printed text value in the receipt.
- “inverse” – Sets printed text to inversed format.
    - true – Printed text is in white font with black background.
    - false – Printed text is in black font and no background.

  Note: This is an optional field.
- “size” – Value for the font size used per line.
    - “default” – Content of line is printed in normal font size.
    - “emphasis” – Content of line is printed in larger than normal font size.
    - “detail” – Content of line is printed in smaller than normal font size.

  Note: This is an optional field.

The following is a sample of a JSON receipt output:

```
“response_jsonrcpt_key”: ”{
    "layout": {
        "format": "condensed",
        "fontmaxchars": [
            {
                "regular": 34
            },
            {
                "emphasis": 24
            },
            {
                "detail": 42
            }
        ]
    },
    "lines": [
        {
            "1": [
                {
                    "type": "line",
                    "align": "c",
                    "size": "emphasis",
                    "font": "bold",
                    "txt": "ADVANCED MOBILE PAYMENT"
                }
            ]
        },
        {
            "2": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": "15 Wertheim Crt."
                }
            ]
        },
        {
            "3": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": "Richmond Hill, ON"
                }
            ]
        },
        {
            "4": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": "L4B 3H7"
                }
            ]
        },
        {
            "5": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": "905-597-2333"
                }
            ]
        },
        {
            "6": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "7": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": "-----------------------------------"
                }
            ]
        },
        {
            "8": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "9": [
                {
                    "type": "line",
                    "align": "c",
                    "size": "emphasis",
                    "font": "bold",
                    "txt": "CREDIT CARD"
                }
            ]
        },
        {
            "10": [
                {
                    "type": "line",
                    "align": "c",
                    "size": "emphasis",
                    "font": "bold",
                    "txt": "Sale"
                }
            ]
        },
        {
            "11": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "12": [
                {
                    "type": "line",
                    "align": "l",
                    "size": "regular",
                    "font": "regular",
                    "txt": "VISA"
                }
            ]
        },
        {
            "13": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "14": [
                {
                    "type": "line",
                    "align": "lr",
                    "size": "default",
                    "font": "regular",
                    "txt1": "************6781",
                    "txt2": "Exp:**/** S"
                }
            ]
        },
        {
            "15": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "16": [
                {
                    "type": "line",
                    "align": "lr",
                    "txt1": "Merchant ID: ",
                    "txt2": "666633340001"
                }
            ]
        },
        {
            "17": [
                {
                    "type": "line",
                    "align": "lr",
                    "txt1": "Invoice #: ",
                    "txt2": "000012"
                }
            ]
```

```
},
{
    "18": [
        {
            "type": "line",
            "align": "lr",
            "txt1": "Trace #: ",
            "txt2": "000200"
        }
    ]
},
{
    "19": [
        {
            "type": "line",
            "align": "c",
            "txt": ""
        }
    ]
},
{
    "20": [
        {
            "type": "line",
            "align": "lr",
            "size": "emphasis",
            "font": "bold",
            "txt1": "Amount",
            "txt2": "CAD$6.00"
        }
    ]
},
{
    "21": [
        {
            "type": "line",
            "align": "c",
            "size": "emphasis",
            "font": "bold",
            "txt": "APPROVED VI0600"
        }
    ]
},
{
    "22": [
        {
            "type": "nextline"
        }
    ]
```

```
},
{
    "23": [
        {
            "type": "line",
            "align": "c",
            "size": "emphasis",
            "font": "bold"
        }
    ]
},
{
    "24": [
        {
            "type": "line",
            "align": "c",
            "size": "emphasis",
            "font": "bold",
            "txt": "00"
        }
    ]
},
{
    "25": [
        {
            "type": "line",
            "align": "lr",
            "txt1": "Gateway TrxnID: ",
            "txt2": "1334061188"
        }
    ]
},
{
    "26": [
        {
            "type": "line",
            "align": "lr",
            "txt1": "Network ID: ",
            "txt2": "010247262007745"
        }
    ]
},
{
    "27": [
        {
            "type": "line",
            "align": "c",
            "txt": ""
        }
```

```
            ]
        },
        {
            "28": [
                {
                    "type": "nextline"
                }
            ]
        },
        {
            "29": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "30": [
                {
                    "type": "line",
                    "align": "c",
                    "size": "detail",
                    "txt": "I AGREE TO PAY ABOVE TOTAL AMOUNT ACCORDING TO CARD ISSUER AGREEMENT (MERCHANT AGREEMENT IF CREDIT VOUCHER)"
                }
            ]
        },
        {
            "31": [
                {
                    "type": "line",
                    "align": "c",
                    "size": "default",
                    "font": "regular",
                    "txt": "Thank You"
                }
            ]
        },
        {
            "32": [
                {
                    "type": "line",
                    "align": "c",
                    "size": "default",
                    "font": "regular",
                    "txt": "Please Come Again"
                }
            ]
        },
        {
            "33": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "34": [
                {
                    "type": "line",
                    "align": "c",
                    "txt": ""
                }
            ]
        },
        {
            "35": [
                {
                    "type": "line",
                    "align": "l",
                    "size": "regular",
                    "font": "regular",
                    "txt": "09/03/2020 18: 02"
                }
            ]
        },
        {
            "36": [
                {
                    "type": "line",
                    "align": "c",
                    "font": "regular",
                    "txt": "MERCHANT COPY"
                }
            ]
        }
    ]
}”
```

Sample message from the AMP Connect:

```
{
    "EndPoint": "TRANSACTION",
    "CmdType": "Purchase",
    "UserDefinedEchoData": "testData",
    "TransactionStatus": "RESULT_OK",
    "Payload": {
        "response_tid_key": "*****001",
        "response_currency_key": "$",
        "response_date_key": "09032020",
        "response_emv_networkid_key": "010247262007745",
        "response_resultcode_key": "0000",
        "response_invoicenum_key": "000012",
        "response_time_key": "180226",
        "response_pan_key": "400300******6781",
        "response_baseamt_key": "600",
        "response_user_defined_echo_data": "testData",
        "response_track1_key": "B400300******6781^V1 VI TEST CARD^**
**502***********************",
        "response_track2_key": "400300******6781=****502************
*",
        "response_totalamt_key": "600",
        "response_issuername": "VISA",
        "response_text_key": "APPROVED",
        "response_code_key": "00",
        "response_card_holder_name_key": "V1 VI TEST CARD",
        "response_authcode_key": "VI0600",
        "response_header1_key": "ADVANCED MOBILE PAYMENT",
        "response_header2_key": "15 Wertheim Crt.",
        "response_header3_key": "Richmond Hill, ON",
        "response_header4_key": "L4B 3H7",
        "response_header5_key": "905-597-2333",
        "response_emv_app_pan_key": "400300******6781",
        "response_mid_key": "*********001",
        "response_cvm_key": "SIGN",
        "response_cardlabel_key": "CREDIT",
        "response_app_version_key": "AMP POS v02.02.022_GPPhu01.00.0
01",
        "response_transname_key": "SALE",
        "response_gatewaytransid": "1334061188",
        "response_trace_key": "000200",
        "response_jsonrcpt_key":"{\n\t\"layout\":\t{\n\t\t\"format\"
:\t\"condensed\",\n\t\t\"fontmaxchars\":\t[{\n\t\t\t\t\"regular\":\t
34\n\t\t\t}, {\n\t\t\t\t\"emphasis\":\t24\n\t\t\t},{\n\t\t\t\t\"deta
il\":\t42\n\t\t\t}]\n\t},\n\t\"lines\":\t[{\n\t\t\t\"1\":\t[{\n\t\t\
t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"align\":\t\"c\",\n\t\t\t\t\t\
"size\":\t\"emphasis\",\n\t\t\t\t\t\"font\":\t\"bold\",\n\t\t\t\t\t\
"txt\":\t\"ADVANCED
MOBILE PAYMENT\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"2\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"15 Wertheim
Crt.\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"3\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"Richmond Hill,
ON\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"4\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"L4B
3H7\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"5\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"905-597-
2333\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"6\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"7\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"--------------------------------
---\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"8\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"9\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"ali
gn\":\t\"c\",\n\t\t\t\t\t\t\"size\":\t\"emphasis\",\n\t\t\t\t\t\t\"font\
":\t\"bold\",\n\t\t\t\t\t\t\"txt\":\t\"CREDIT
CARD\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"10\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\t\"size\":\t\"emphasis\",\n\t\t\t\t\t\t\"font
\":\t\"bold\",\n\t\t\t\t\t\t\"txt\":\t\"Sale\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"11\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"12\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"l\",\n\t\t\t\t\t\t\"size\":\t\"regular\",\n\t\t\t\t\t\t\"font\
":\t\"regular\",\n\t\t\t\t\t\t\"txt\":\t\"VISA\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"13\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"14\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\t\"size\":\t\"default\",\n\t\t\t\t\t\t\"font
\":\t\"regular\",\n\t\t\t\t\t\t\"txt1\":\t\"************6781\",\n\t\t\
t\t\t\t\"txt2\":\t\"Exp:
**\/** S\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"15\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"16\":\t[{\n\t\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\t\"txt1\":\t\"Merchant
ID:\",\n\t\t\t\t\t\t\"txt2\":\t\"666633340001\"\n\t\t\t\t\t}]\n\t\t},
{\n\t\t\t\"17\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\"txt1\":\t\"Invoice
#:\",\n\t\t\t\t\t\"txt2\":\t\"000012\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"18\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\"txt1\":\t\"Trace
#:\",\n\t\t\t\t\t\"txt2\":\t\"000200\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"19\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"20\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\"size\":\t\"emphasis\",\n\t\t\t\t\t\"fon
t\":\t\"bold\",\n\t\t\t\t\t\"txt1\":\t\"Amount\",\n\t\t\t\t\t\"txt2\
":\t\"CAD$
6.00\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"21\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"size\":\t\"emphasis\",\n\t\t\t\t\t\"font
\":\t\"bold\",\n\t\t\t\t\t\"txt\":\t\"APPROVED
VI0600\"\n\t\t\t\t}]\n\t\t}, {\n\t\t\t\"22\":\t[{\n\t\t\t\t\t\"type\
":\t\"nextline\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"23\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"size\":\t\"emphasis\",\n\t\t\t\t\t\"font
\":\t\"bold\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"24\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"size\":\t\"emphasis\",\n\t\t\t\t\t\"font
\":\t\"bold\",\n\t\t\t\t\t\"txt\":\t\"00\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"25\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\"txt1\":\t\"Gateway Trxn
ID:\",\n\t\t\t\t\t\"txt2\":\t\"1334061188\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"26\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"lr\",\n\t\t\t\t\t\"txt1\":\t\"Network
ID:\",\n\t\t\t\t\t\"txt2\":\t\"010247262007745\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"27\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t}]\n\t\t}, {\n\t\t\t\"28\":\t[{\n\t\t\t\t\t\"type\":\t\"
nextline\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"29\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"30\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"size\":\t\"detail\",\n\t\t\t\t\t\"txt\":
\t\"I
AGREE TO PAY ABOVE TOTAL AMOUNT ACCORDING TO CARD ISSUER AGREEMENT (
MERCHANT AGREEMENT IF CREDIT
VOUCHER)\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"31\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"size\":\t\"default\",\n\t\t\t\t\t\"font\
":\t\"regular\",\n\t\t\t\t\t\"txt\":\t\"Thank
You\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"32\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"size\":\t\"default\",\n\t\t\t\t\t\"font\
":\t\"regular\",\n\t\t\t\t\t\"txt\":\t\"Please
Come Again\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"33\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"34\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"txt\":\t\"
\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"35\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"l\",\n\t\t\t\t\t\"size\":\t\"regular\",\n\t\t\t\t\t\"font\
":\t\"regular\",\n\t\t\t\t\t\"txt\":\t\"09\/03\/2020
18:02\"\n\t\t\t\t}]\n\t\t},
{\n\t\t\t\"36\":\t[{\n\t\t\t\t\t\"type\":\t\"line\",\n\t\t\t\t\t\"al
ign\":\t\"c\",\n\t\t\t\t\t\"font\":\t\"regular\",\n\t\t\t\t\t\"txt\"
:\t\"MERCHANT
COPY\"\n\t\t\t\t}]\n\t\t}]\n}"}}
```

APPENDIX A: RESPONSE TAGS

RESPONSE TAGS RETURN FROM TRANSACTION ENDPOINT BY THE PAYMENT APPLICATION

The following tags are returned by the AMP payment application, in "Payload" tag, when the requested command is processed.

Response Tag	Description
response_merch_name_key	Merchant Name
response_merch_add1_key	Merchant Address 1
response_merch_add2_key	Merchant Address 2
response_cust_serv_phone_key	Customer Service Phone
response_time_key	Transaction Time
response_date_key	Transaction Date
response_tid_key	Terminal ID
response_mid_key	Merchant ID
response_transname_key	Transaction Name
response_card_holder_name_key	Cardholder Name
response_cardlabel_key	Card Label
response_pan_key	Masked Primary Account Number (only showing the first six and last four digits)
response_track1_key	Masked Track1 data of the card
response_track2_key	Masked Track2 data of the card
response_expirydate_key	Card Expiration Date
response_entrymode_key	Card Entry Mode
response_cvm_key	Card Verification Method
response_invoicenum_key	Invoice Number

response_clerkid_key	Clerk ID
response_storenum_key	Store Number
response_trace_key	Trace Number
response_batch_key	Batch Number
response_user_defined_echo_data	Unique ID from Host identifying the transaction
response_baseamt_key	Base Amount
response_tipamt_key	Tip Amount
response_cashbkamt_key	Cashback Amount
response_surchargeamt_key	Surcharge Amount
response_reversal_amt_key	Reversed Amount
response_totalamt_key	Total Amount
response_tvr_key	Terminal Verification Result
response_aid_key	Application Identifier
response_authcode_key	Authorization Code
response_sequencenum_key	Sequence Number
response_currency_key	Currency
response_iad_key	Issuer Application Data
response_tsi_key	Transaction Status Indicator
response_arc_key	Authorisation Response Code
response_code_key	Response Code from Host
response_text_key	Response Text from Host
response_appprefname_key	Application Preferred Name
response_tc_key	Transaction Certificate
response_hosttimestamp_key	Host Timestamp
response_transref_key	Transaction Reference Number

response_xmldata_key	Certain transactions and operations utilize this response key to return multiple values in XML format.
response_tagvalue_key	The value obtained from the configuration file of BaseApp.
response_resultcode_key	Transaction result code.
response_header1_key	Header Line 1 of the receipt
response_header2_key	Header Line 2 of the receipt
response_header3_key	Header Line 3 of the receipt
response_header4_key	Header Line 4 of the receipt
response_header5_key	Header Line 5 of the receipt
response_header6_key	Header Line 6 of the receipt
response_footer1_key	Footer Line 1 of the receipt
response_footer2_key	Footer Line 2 of the receipt
response_footer3_key	Footer Line 3 of the receipt
response_footer4_key	Footer Line 4 of the receipt
response_footer5_key	Footer Line 5 of the receipt
response_footer6_key	Footer Line 6 of the receipt
response_transid_key	Batch Transaction ID
response_cashbackfee_key	Cashback Fee
response_noncashfee_key	Non-cash Fee
response_emv_settledate_key	Settlement Date
response_emv_networkid_key	Network ID
response_emv_applabel_key	EMV Application Label
response_emv_app_pan_key	EMV Application Primary Account Number
response_emv_currencycode_key	EMV Currency Code

response_emv_panseqnum_key	EMV PAN Sequence Number
response_emv_aip_key	EMV Application Interchange Profile
response_emv_tvr_key	EMV Transaction Verification Results
response_emv_transdate_key	EMV Transaction Date
response_emv_transtype_key	EMV Transaction Type
response_emv_auth_amt_key	EMV Authorized Amount
response_emv_oth_amt_key	EMV Other Amount
response_emv_usage_ctrl_key	EMV Usage Control
response_emv_iac_online_key	EMV Issuer Action Code – Online
response_emv_iac_default_key	EMV Issuer Action Code – Default
response_emv_iac_denial_key	EMV Issuer Action Code – Denial
response_emv_termcode_key	EMV Terminal Country code
response_emv_ac_key	EMV Application Cryptogram
response_emv_cid_key	EMV Cryptogram Information data
response_emv_iad_key	EMV Issuer Application Data
response_emv_cvm_key	EMV Card Verification Mode
response_emv_atc_key	EMV Application Transaction Counter
response_emv_unpred_num_key	EMV Unpredictable Number
response_emv_tac_online_key	EMV Terminal Action Code - Online
response_emv_tac_default_key	EMV Terminal Action Code - Default
response_emv_tac_denial_key	EMV Terminal Action Code - Denial
response_token_data	Tokenized Data
response_app_version_key	Application Version
response_emv_hosttimestamp_key	EMV Host timestamp
response_emv_app_transdata	EMV Application Transaction Data
response_code_iso_key	ISO Response Code

response_swver_key	PayFacto Software Version assigned key
response_issuername	Issuer Name
response_gatewaytransid	Gateway Transaction ID
response_bankresponsecode	Bank Response Code
response_available_amt_key	Available Amount
Response_additionalData_Key	Return the same data send in additional data tag in request transaction

APPENDIX B: SAMPLE DISCOVERY CODE

The following are sample codes to demonstrate the discovery to find the AMP Connect IP.

ANDROID

1) Create NsdManager instance and discovery nsd services. The result will be delivered to the listener.

```
NsdManager nsdManager = (NsdManager)
getSystemService(Context.NSD_SERVICE);
nsdManager.discoverServices("_http._tcp.",
NsdManager.PROTOCOL_DNS_SD, discoveryListener);
```

2) Define the Listener.

```
    NsdManager.DiscoveryListener discoveryListener = new
NsdManager.DiscoveryListener() {
        @Override
        public void onStartDiscoveryFailed(String serviceType, int
errorCode) {
            if (nsdManager != null) {
                nsdManager.stopServiceDiscovery(this);
            }
        }

        @Override
        public void onStopDiscoveryFailed(String serviceType, int
errorCode) {
            if (nsdManager != null) {
                nsdManager.stopServiceDiscovery(this);
            }
        }

        @Override
        public void onDiscoveryStarted(String serviceType) {
        }

        @Override
        public void onDiscoveryStopped(String serviceType) {
        }

        @Override
        public void onServiceFound(NsdServiceInfo serviceInfo) {
            if (isDesiredService(serviceInfo)) {   //Check if the
found service is desired, go 4
```

```
                try {
                    nsdManager.resolveService(serviceInfo,
resolveListener);   // Resolve the found listener, go 5
                } catch (Exception e) {
                    Log.e(TAG, e.getMessage());
                }
            }
        }

        @Override
        public void onServiceLost(NsdServiceInfo serviceInfo) {
        }
    };
```

3) Check if the found service is the desired one.

```
private boolean isDesiredService(NsdServiceInfo serviceInfo) {
    if (nsdManager == null || serviceInfo == null) {
            return false;
        }

        String serviceName = serviceInfo.getServiceName();
        if (serviceName == null || serviceName.length() == 0) {
            return false;
        }

    return serviceName.equals(AMPConnectBroadcastServiceName);
        //AMPConnectBroadcastServiceName should be your device
serial number
}
```

4) Define resolveListener.

```
    NsdManager.ResolveListener resolveListener = new
NsdManager.ResolveListener() {
        @Override
        public void onResolveFailed(NsdServiceInfo serviceInfo, int
errorCode) {
        }

        @Override
        public void onServiceResolved(NsdServiceInfo serviceInfo) {
        // NsdServiceInfo received
            if (isDesiredService(serviceInfo)) { // Check again
                    String AMPConnectTerminalIp =
serviceInfo.getHost().getHostAddress();
            }
        }
    };
```

5) The IP address of the terminal that runs AMP Connect is determined. Build the URL.

```
private String getUrl(EndPoint endPoint) {
        return "http://" + AMPConnectTerminalIp  + ":22222" + "/" +
endPoint.name();
   }
```

Use the determined URL to communicate with the AMP Connect app.

WINDOWS C#

```
using System;
using System.Collections.Generic;
using Windows.Devices.Enumeration;
using Windows.Networking;

namespace AMPConnectInterface
{
    class CatchServiceInfo
    {

        private static Guid DnsSdProtocol = new Guid("{4526e8c1-
8aac-4153-9b16-55e86ada0e54}");

        private string _queryString =
"System.Devices.AepService.ProtocolId:={" + DnsSdProtocol + "} AND "
+
            "System.Devices.Dnssd.Domain:=\"local\" AND
System.Devices.Dnssd.ServiceName:=\"_http._tcp\" AND
System.Devices.Dnssd.InstanceName:=\"";

        private const string GET_TERMINAL_INFO =
"GET_TERMINAL_INFO";

        private DeviceWatcher _watcher;

        private const string HOSTNAME_PROPERTY =
"System.Devices.Dnssd.HostName";
        private const string IPADDRESS_PROPERTY =
"System.Devices.IpAddress";
        private const string PORTNUMBER_PROPERTY =
"System.Devices.Dnssd.PortNumber";
```

```
        private const string SERVICENAME_PROPERTY = 
"System.Devices.Dnssd.ServiceName";
        private const string INSTANCENAME_PROPERTY = 
"System.Devices.Dnssd.InstanceName";

        private string _response = "";
        private bool _isResponseAvailabel;

        private string[] _propertyKeys = new String[] {
            HOSTNAME_PROPERTY,
            SERVICENAME_PROPERTY,
            INSTANCENAME_PROPERTY,
            IPADDRESS_PROPERTY,
            PORTNUMBER_PROPERTY
        };

        public CatchServiceInfo(string _instanceName)
        {
            _queryString += _instanceName + "\"";

            _watcher = DeviceInformation.CreateWatcher(
                       _queryString,
                       _propertyKeys,

DeviceInformationKind.AssociationEndpointService);

            _watcher.Added += (s, a) => 
ConnectToService(a.Properties);
            _watcher.Start();
        }

        public string GetResponse()
        {
            return _response;
        }

        public bool IsResponseReady()
        {
            return _isResponseAvailabel;
        }

        ~CatchServiceInfo()
        {
            _watcher.Stop();
        }

        void ConnectToService(IReadOnlyDictionary<string, object> 
properties)
        {
            var host = new HostName((properties[IPADDRESS_PROPERTY]
as String[])[0]);
            var port = properties[PORTNUMBER_PROPERTY].ToString();

            _isResponseAvailabel = TestConnection(host, port);

        }

        bool TestConnection(HostName host, string port)
        {
            try
            {
                //http://192.168.0.14:22222/GET_TERMINAL_INFO

                RestClient restClient = new RestClient("http://" +
host.ToString() + ":" + port + "/" + GET_TERMINAL_INFO);
                restClient.httpMethod = HttpVerb.POST;
                _response = restClient.MakeRequest();
            }
            catch (Exception)
            {
                return false;
            }

            return true;
        }

    }
```

APPENDIX C: SAMPLE MESSAGE

SAMPLE MESSAGE STEPS, REQUEST AND RESPONSE SAME NETWORK

- Request for a transaction:

```
{
"cmdType":"Purchase",
"ReqPayload":{
"BaseAmount":"100",
"UserDefinedEchoData":"testData",
"CardEntryMethod":"AUTO",
“autoPrint”:”TRUE”
}
```

- Response:

```
{"EndPoint":"TRANSACTION",
"CmdType":"Purchase",
"UserDefinedEchoData":"testData",
"TransactionStatus":"RESULT_OK",
"Payload":{
"response_tid_key":"AMP00025",
"response_currency_key":"$","response_date_key":"200421","response_r
esultcode_key":"0000","response_tsi_key":"6800","response_aid_key":"
A0000002771010","response_tvr_key":"8080008000","response_invoicenum
_key":"0000025","response_time_key":"100758","response_pan_key":"589
297*********1014","response_arc_key":"","response_cashbkamt_key":"",
"response_storenum_key":"","response_baseamt_key":"10000","response_
user_defined_echo_data":"testData","response_code_iso_key":"00","res
ponse_track1_key":null,"response_track2_key":"589297*********1014=**
**220*********","response_totalamt_key":"10000","response_appprefnam
e_key":"Interac","response_text_key":"TRANSACTION
APPROVED","response_code_key":"000","response_sequencenum_key":"1001
121","response_card_holder_name_key":"Test Card
1","response_clerkid_key":"","response_authcode_key":"481406","respo
nse_footer1_key":"","response_footer2_key":"","response_footer3_key"
:"","response_footer4_key":"","response_footer5_key":"","response_fo
oter6_key":"","response_header1_key":"AMP TEMP TERMINAL
TESTING","response_header2_key":"15 WERTHEIM CRT. UNITS 401-
403","response_header3_key":"RICHMOND HILL ON
L4B3H7","response_header4_key":"AMP TESTING KRISTENE
CONCHA","response_header5_key":"","response_header6_key":"","respons
e_entrymode_key":"Dipped","response_batch_key":"66","response_mid_ke
y":"AMP0000001","response_cvm_key":"PIN
VERIFIED","response_cardlabel_key":"DEBIT/SAV","response_iad_key":"0
6040A03A40000","response_tc_key":"1B6A831C2A5E3DAE","response_revers
al_amt_key":"","response_cust_serv_phone_key":"","response_swver_key
":"AMP_AMP_8SER_V100C","response_transref_key":"","response_tipamt_k
ey":"","response_tagvalue_key":"","response_hosttimestamp_key":"04/2
1/20 10:08:29","response_app_version_key":"AMP POS
v02.02.020PFPa","response_surchargeamt_key":"","response_transname_k
ey":"Purchase","response_trace_key":null,"response_xmldata_key":""
}
}
```

USB OR SERIAL

- Request:

```
{
"Endpoint":"TRANSACTION","cmdType":"Purchase",
"ReqPayload":{
"BaseAmount":"100",
"UserDefinedEchoData":"testData",
"CardEntryMethod":"AUTO",
“autoPrint”:”TRUE”
}
```

- Response:

```
{"EndPoint":"TRANSACTION",
"CmdType":"Purchase",
"UserDefinedEchoData":"testData",
"TransactionStatus":"RESULT_OK",
"Payload":{
"response_tid_key":"AMP00025",
"response_currency_key":"$","response_date_key":"200421","response_r
esultcode_key":"0000","response_tsi_key":"6800","response_aid_key":"
A0000002771010","response_tvr_key":"8080008000","response_invoicenum
_key":"0000025","response_time_key":"100758","response_pan_key":"589
297*********1014","response_arc_key":"","response_cashbkamt_key":"",
"response_storenum_key":"","response_baseamt_key":"10000","response_
user_defined_echo_data":"testData","response_code_iso_key":"00","res
ponse_track1_key":null,"response_track2_key":"589297*********1014=**
**220*********","response_totalamt_key":"10000","response_appprefnam
e_key":"Interac","response_text_key":"TRANSACTION
APPROVED","response_code_key":"000","response_sequencenum_key":"1001
121","response_card_holder_name_key":"Test Card
1","response_clerkid_key":"","response_authcode_key":"481406","respo
nse_footer1_key":"","response_footer2_key":"","response_footer3_key"
:"","response_footer4_key":"","response_footer5_key":"","response_fo
oter6_key":"","response_header1_key":"AMP TEMP TERMINAL
TESTING","response_header2_key":"15 WERTHEIM CRT. UNITS 401-
403","response_header3_key":"RICHMOND HILL ON
L4B3H7","response_header4_key":"AMP TESTING KRISTENE
CONCHA","response_header5_key":"","response_header6_key":"","respons
e_entrymode_key":"Dipped","response_batch_key":"66","response_mid_ke
y":"AMP0000001","response_cvm_key":"PIN
VERIFIED","response_cardlabel_key":"DEBIT/SAV","response_iad_key":"0
6040A03A40000","response_tc_key":"1B6A831C2A5E3DAE","response_revers
al_amt_key":"","response_cust_serv_phone_key":"","response_swver_key
":"AMP_AMP_8SER_V100C","response_transref_key":"","response_tipamt_k
ey":"","response_tagvalue_key":"","response_hosttimestamp_key":"04/2
1/20 10:08:29","response_app_version_key":"AMP POS
v02.02.020PFPa","response_surchargeamt_key":"","response_transname_k
ey":"Purchase","response_trace_key":null,"response_xmldata_key":""
}
}
```

CLOUD

- Request:

```
{
"sourceSerial": "1234567890",
"targetTerminal": "8000021982",
"authCode": "9845z77t",
"requestData":"
{
"Endpoint":"TRANSACTION","cmdType":"Purchase",
"ReqPayload":{
"BaseAmount":"100",
"UserDefinedEchoData":"testData",
"CardEntryMethod":"AUTO",
“autoPrint”:”TRUE”
}
}"
```

- Response from Status Command:

```
{"canReload":true,
"result":{
"id":"bd418e67-9b29-438e-90fd-fe945bd1380b",
"sourceSerial":"1234567890",
"targetTerminal":"8000021982",
"requestData":"
{
"Endpoint":"TRANSACTION","cmdType":"Purchase",
"ReqPayload":{
"BaseAmount":"100",
"UserDefinedEchoData":"testData",
"CardEntryMethod":"AUTO",
“autoPrint”:”TRUE”
}
}",
"responseData":"{
"EndPoint":"TRANSACTION",
"CmdType":"Purchase",
"UserDefinedEchoData":"testData",
"TransactionStatus":"RESULT_OK",
"Payload":{
"response_tid_key":"AMP00025",
"response_currency_key":"$","response_date_key":"200421","response_r
esultcode_key":"0000","response_tsi_key":"6800","response_aid_key":"
A0000002771010","response_tvr_key":"8080008000","response_invoicenum
_key":"0000025","response_time_key":"100758","response_pan_key":"589
297*********1014","response_arc_key":"","response_cashbkamt_key":"",
"response_storenum_key":"","response_baseamt_key":"10000","response_
user_defined_echo_data":"testData","response_code_iso_key":"00","res
ponse_track1_key":null,"response_track2_key":"589297*********1014=**
**220*********","response_totalamt_key":"10000","response_appprefnam
e_key":"Interac","response_text_key":"TRANSACTION
APPROVED","response_code_key":"000","response_sequencenum_key":"1001
121","response_card_holder_name_key":"Test Card
1","response_clerkid_key":"","response_authcode_key":"481406","respo
nse_footer1_key":"","response_footer2_key":"","response_footer3_key"
:"","response_footer4_key":"","response_footer5_key":"","response_fo
oter6_key":"","response_header1_key":"AMP TEMP TERMINAL
TESTING","response_header2_key":"15 WERTHEIM CRT. UNITS 401-
403","response_header3_key":"RICHMOND HILL ON
L4B3H7","response_header4_key":"AMP TESTING KRISTENE
CONCHA","response_header5_key":"","response_header6_key":"","respons
e_entrymode_key":"Dipped","response_batch_key":"66","response_mid_ke
y":"AMP0000001","response_cvm_key":"PIN
VERIFIED","response_cardlabel_key":"DEBIT/SAV","response_iad_key":"0
6040A03A40000","response_tc_key":"1B6A831C2A5E3DAE","response_revers
al_amt_key":"","response_cust_serv_phone_key":"","response_swver_key
":"AMP_AMP_8SER_V100C","response_transref_key":"","response_tipamt_k
ey":"","response_tagvalue_key":"","response_hosttimestamp_key":"04/2
1/20 10:08:29","response_app_version_key":"AMP POS
v02.02.020PFPa","response_surchargeamt_key":"","response_transname_k
ey":"Purchase","response_trace_key":null,"response_xmldata_key":""
}
}",
"status":"DONE","createdAt":"2020-04-
21T14:07:56.000Z","updatedAt":"2020-04-
21T14:28:11.000Z","TerminalId":"8bb60419-5e50-4937-b3b2-
5a2f1224a390"}
,"message":"Successfully found the record."
}
```

APPENDIX D: SAMPLE PROGRAMMING STEPS

USB/SERIAL

Send the transaction to AMP Connect after connecting to the terminal, using a USB or a serial cable. Make sure that you set the right baud rate and configuration (8N1, 115200).

```
{“EndPoint”:”TRANSACTION”,
"cmdType": "Purchase",
        "ReqPayload": {
            "BaseAmount": "3.4",
            "UserDefinedEchoData": "testData",
            "CardEntryMethod": "AUTO",
            "AutoPrint": "true"
        }
 }
```

1) The response that is returned by the AMP Connect is in the below format. The business app should verify that the response is “0”, which means that the transaction is accepted:

```
{"AMPConnectResponseCode":0}
```

Refer to **AMP Connect Response Code** for details.

The business application should check the result by sending the following command until the TransactionStatus":"RESULT_OK" is received.

```
{“EndPoint”:”TRANSACTION”,
"cmdType":"GetTransactionResult"}
```

The transaction is properly executed when the "response_resultcode_key":"0000”. The rest of the tags should be evaluated based on **BaseApp Response Result Codes.**

```
{"EndPoint":"TRANSACTION","CmdType":"Purchase","UserDefinedEchoData"
:"testData","TransactionStatus":"RESULT_OK","Payload":{"response_tid
_key":"AMP00023","response_currency_key":"$","response_date_key":"20
0424","response_resultcode_key":"0000","response_tsi_key":"6800","re
sponse_aid_key":"A0000000031010","response_tvr_key":"8080008000","re
sponse_invoicenum_key":"0000006","response_time_key":"111808","respo
nse_pan_key":"452070******8023","response_arc_key":"","response_cash
bkamt_key":"","response_storenum_key":"","response_baseamt_key":"340
","response_user_defined_echo_data":"testData","response_code_iso_ke
y":"","response_track1_key":"B452070******8023^TESTCARD
\/H
^****201************************","response_track2_key":"452070*****
*8023=****201*************","response_totalamt_key":"340","response_
appprefname_key":"Visa
Credit","response_text_key":"Transaction Not
Completed","response_code_key":"366","response_sequencenum_key":"100
3726","response_card_holder_name_key":"TEST \/H
","response_clerkid_key":"","response_authcode_key":"","response_foo
ter1_key":"","response_footer2_key":"","response_footer3_key":"","re
sponse_footer4_key":"","response_footer5_key":"","response_footer6_k
ey":"","response_header1_key":"","response_header2_key":"","response
_header3_key":"","response_header4_key":"","response_header5_key":""
,"response_header6_key":"","response_entrymode_key":"Fallback
Swipe","response_batch_key":"290","response_mid_key":"AMP0000001","r
esponse_cvm_key":"SIGN","response_cardlabel_key":"CREDIT\/VISA","res
ponse_iad_key":"06010A03A08000","response_tc_key":"C752496D3FED645A"
,"response_reversal_amt_key":"","response_cust_serv_phone_key":"","r
esponse_swver_key":"AMP_AMP_8SER_V100C","response_transref_key":"011
515523801","response_tipamt_key":"","response_tagvalue_key":"","resp
onse_hosttimestamp_key":"04\/24\/20
11:18:08","response_app_version_key":"AMP POS
v02.02.020PFPa","response_surchargeamt_key":"","response_transname_k
ey":"Purchase","response_trace_key":null,"response_xmldata_key":""}}
```

SAME NETWORK

1) Send the transaction to AMP Connect after the terminal’s IP address discovery. If the IP address is static in the same network, the static IP address can be used.

```
POST>http://192.168.1.196:22222/TRANSACTION
{"cmdType": "Purchase",
        "ReqPayload": {
            "BaseAmount": "3.4",
            "UserDefinedEchoData": "testData",
            "CardEntryMethod": "AUTO",
            "AutoPrint": "true"
        }
 }
```

The response that is returned by the AMP Connect is in the below format. The business app should verify that the response is “0”, which means that the transaction is accepted:

```
{"AMPConnectResponseCode":0}
```

Refer to **AMP Connect Response Code** for details.

2) The business application should check the result by sending the following command until the TransactionStatus":"RESULT_OK" is received.

```
POST>http://192.168.1.196:22222/TRANSACTION
{"cmdType":"GetTransactionResult"}
```

The transaction is properly executed when the "response_resultcode_key":"0000”. The rest of tags should be evaluated based on **BaseApp Response Result Codes.**

```
{"EndPoint":"TRANSACTION","CmdType":"Purchase","UserDefinedEchoData"
:"testData","TransactionStatus":"RESULT_OK","Payload":{"response_tid
_key":"AMP00023","response_currency_key":"$","response_date_key":"20
0424","response_resultcode_key":"0000","response_tsi_key":"6800","re
sponse_aid_key":"A0000000031010","response_tvr_key":"8080008000","re
sponse_invoicenum_key":"0000006","response_time_key":"111808","respo
nse_pan_key":"452070******8023","response_arc_key":"","response_cash
bkamt_key":"","response_storenum_key":"","response_baseamt_key":"340
","response_user_defined_echo_data":"testData","response_code_iso_ke
y":"","response_track1_key":"B452070******8023^TESTCARD
\/H
^****201***********************","response_track2_key":"452070*****
*8023=****201************","response_totalamt_key":"340","response_
appprefname_key":"Visa
Credit","response_text_key":"Transaction Not
Completed","response_code_key":"366","response_sequencenum_key":"100
3726","response_card_holder_name_key":"TEST \/H
","response_clerkid_key":"","response_authcode_key":"","response_foo
ter1_key":"","response_footer2_key":"","response_footer3_key":"","re
sponse_footer4_key":"","response_footer5_key":"","response_footer6_k
ey":"","response_header1_key":"","response_header2_key":"","response
_header3_key":"","response_header4_key":"","response_header5_key":""
,"response_header6_key":"","response_entrymode_key":"Fallback
Swipe","response_batch_key":"290","response_mid_key":"AMP0000001","r
esponse_cvm_key":"SIGN","response_cardlabel_key":"CREDIT\/VISA","res
ponse_iad_key":"06010A03A08000","response_tc_key":"C752496D3FED645A"
,"response_reversal_amt_key":"","response_cust_serv_phone_key":"","r
esponse_swver_key":"AMP_AMP_8SER_V100C","response_transref_key":"011
515523801","response_tipamt_key":"","response_tagvalue_key":"","resp
onse_hosttimestamp_key":"04\/24\/20
11:18:08","response_app_version_key":"AMP POS
v02.02.020PFPa","response_surchargeamt_key":"","response_transname_k
ey":"Purchase","response_trace_key":null,"response_xmldata_key":""}}
```

CLOUD

1) Send the transaction to the AMP Cloud.

- Request:

```
POST>https://ecr.amobilepayment.com:3000/api/command/queue/
{
     "sourceSerial": "1234567890",
     "targetTerminal": "8200000562",
,     "authCode": "viyuji48"
     "payload": {
         "Endpoint": "TRANSACTION",
         "cmdType": "Purchase",
"ReqPayload": {
             "BaseAmount": "3.4",
             "UserDefinedEchoData": "testData",
             "CardEntryMethod": "AUTO",
             "AutoPrint": "true"
         }}}
```

The response includes the commandId, which is unique per transaction, and should be used for any future enquiries.

The "canReload": true means the command is executed successfully by the AMP Cloud.

- Response:

```
{
    "canReload": “true”,
    "result": {
        "commandId": "8ff28c53-65ac-443a-a0d7-efe4df5bcd4d"
    },
    "message": "Successfully queued the command."
}
```

2) The business application should check the result by sending the following command until “canReload":”true” and “status”:”DONE” or ”RECEIVED_RESPONSE”.

The transaction is executed properly when the "response_resultcode_key":"0000” and TransactionStatus":"RESULT_OK. The rest of tags should be evaluated based **BaseApp Response Result Codes** and **AMP Connect Response Code**.

- Request:

```
https://ecr.amobilepayment.com:3000/api/command/response/8ff28c53-
65ac-443a-a0d7-efe4df5bcd4d
```

- Response:

```
{"canReload":true,"result":{"EndPoint":"TRANSACTION","CmdType":"Purc
hase","UserDefinedEchoData":"testData","TransactionStatus":"RESULT_O
K","Payload":{"response_tid_key":"AMP00023","response_currency_key":
"$","response_date_key":"200424","response_resultcode_key":"0000","r
esponse_tsi_key":"","response_aid_key":"","response_tvr_key":"","res
ponse_invoicenum_key":"0000009","response_time_key":"115117","respon
se_pan_key":"452070******8023","response_arc_key":"","response_cashb
kamt_key":"","response_storenum_key":"","response_baseamt_key":"340"
,"response_user_defined_echo_data":"testData","response_code_iso_key
":"","response_track1_key":"B452070******8023^TESTCARD
/H
^****201************************","response_track2_key":"452070*****
*8023=****201************","response_totalamt_key":"340","response_
appprefname_key":"","response_text_key":"Transaction Not
Completed","response_code_key":"366","response_sequencenum_key":"100
3727","response_card_holder_name_key":"TESTCARD             /H
","response_clerkid_key":"","response_authcode_key":"","response_foo
ter1_key":"","response_footer2_key":"","response_footer3_key":"","re
sponse_footer4_key":"","response_footer5_key":"","response_footer6_k
ey":"","response_header1_key":"","response_header2_key":"","response
_header3_key":"","response_header4_key":"","response_header5_key":""
,"response_header6_key":"","response_entrymode_key":"Fallback
Swipe","response_batch_key":"290","response_mid_key":"AMP0000001","r
esponse_cvm_key":"SIGN","response_cardlabel_key":"CREDIT/VISA","resp
onse_iad_key":"","response_tc_key":"","response_reversal_amt_key":""
,"response_cust_serv_phone_key":"","response_swver_key":"AMP_AMP_8SE
R_V100C","response_transref_key":"011515524003","response_tipamt_key
":"","response_tagvalue_key":"","response_hosttimestamp_key":"04/24/
20 11:51:18","response_app_version_key":"AMP POS
v02.02.020PFPa","response_surchargeamt_key":"","response_transname_k
ey":"Purchase","response_trace_key":null,"response_xmldata_key":""}}
,"message":"Successfully fetched the command response."}
```

APPENDIX E: POWER FAILURE RECOVERY

POWER FAILURE OR LOST CONNECTION RECOVERY SCENARIOS

When the business application calls the payment application transactions, power or connection failure can happen between or during the transactions. The business application should be able to automatically recover, specially in an unattended environment.

The following are the identified scenarios and the recommended recovery methods.

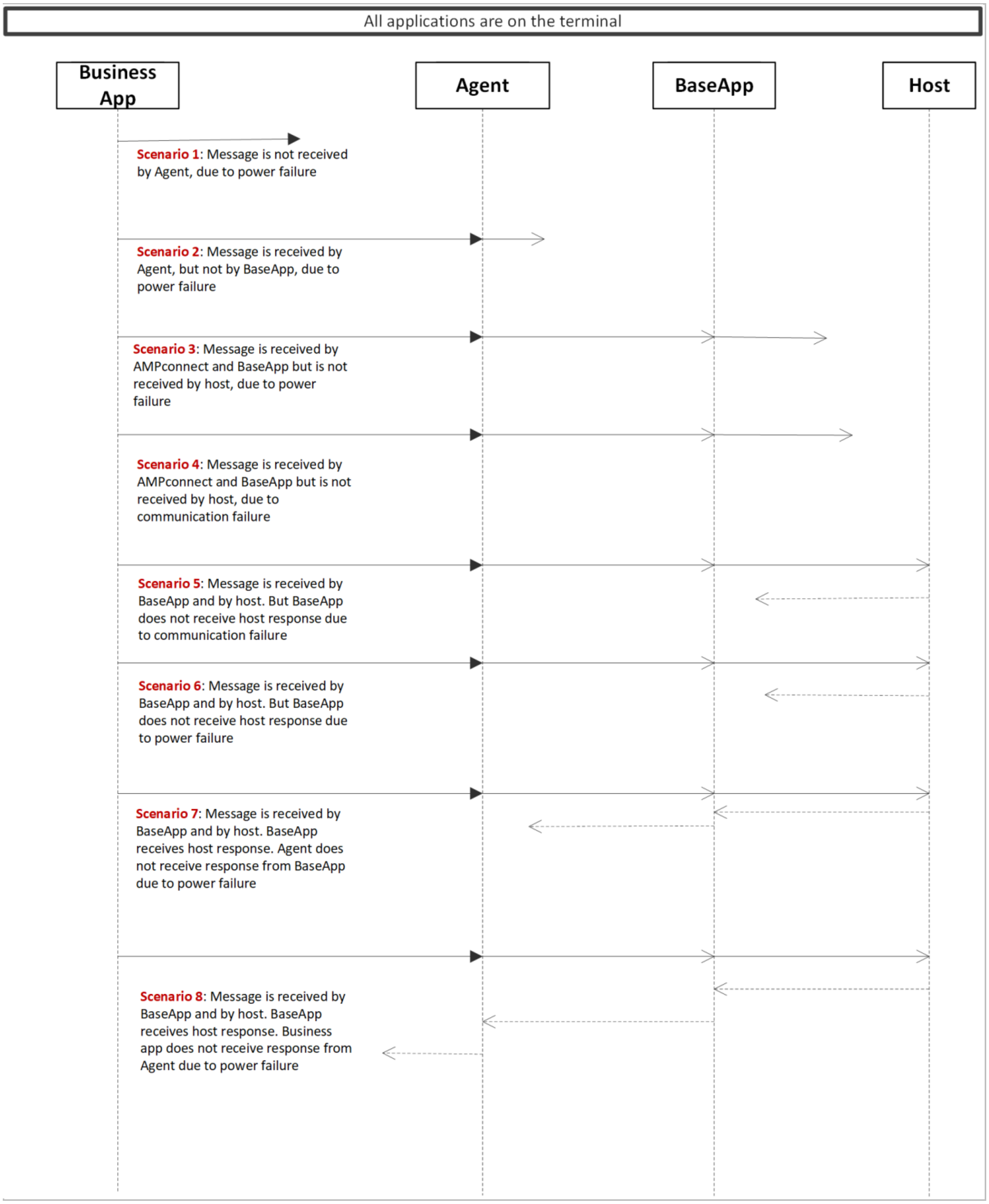

All applications are on the terminal
Business App
Agent
BaseApp
Host
Scenario 1: Message is not received by Agent, due to power failure
Scenario 2: Message is received by Agent, but not by BaseApp, due to power failure
Scenario 3: Message is received by AMPconnect and BaseApp but is not received by host, due to power failure
Scenario 4: Message is received by AMPconnect and BaseApp but is not received by host, due to communication failure
Scenario 5: Message is received by BaseApp and by host. But BaseApp does not receive host response due to communication failure
Scenario 6: Message is received by BaseApp and by host. But BaseApp does not receive host response due to power failure
Scenario 7: Message is received by BaseApp and by host. BaseApp receives host response. Agent does not receive response from BaseApp due to power failure
Scenario 8: Message is received by BaseApp and by host. BaseApp receives host response. Business app does not receive response from Agent due to power failure

Figure 10 - Recovery when All Apps Are on the Terminal.

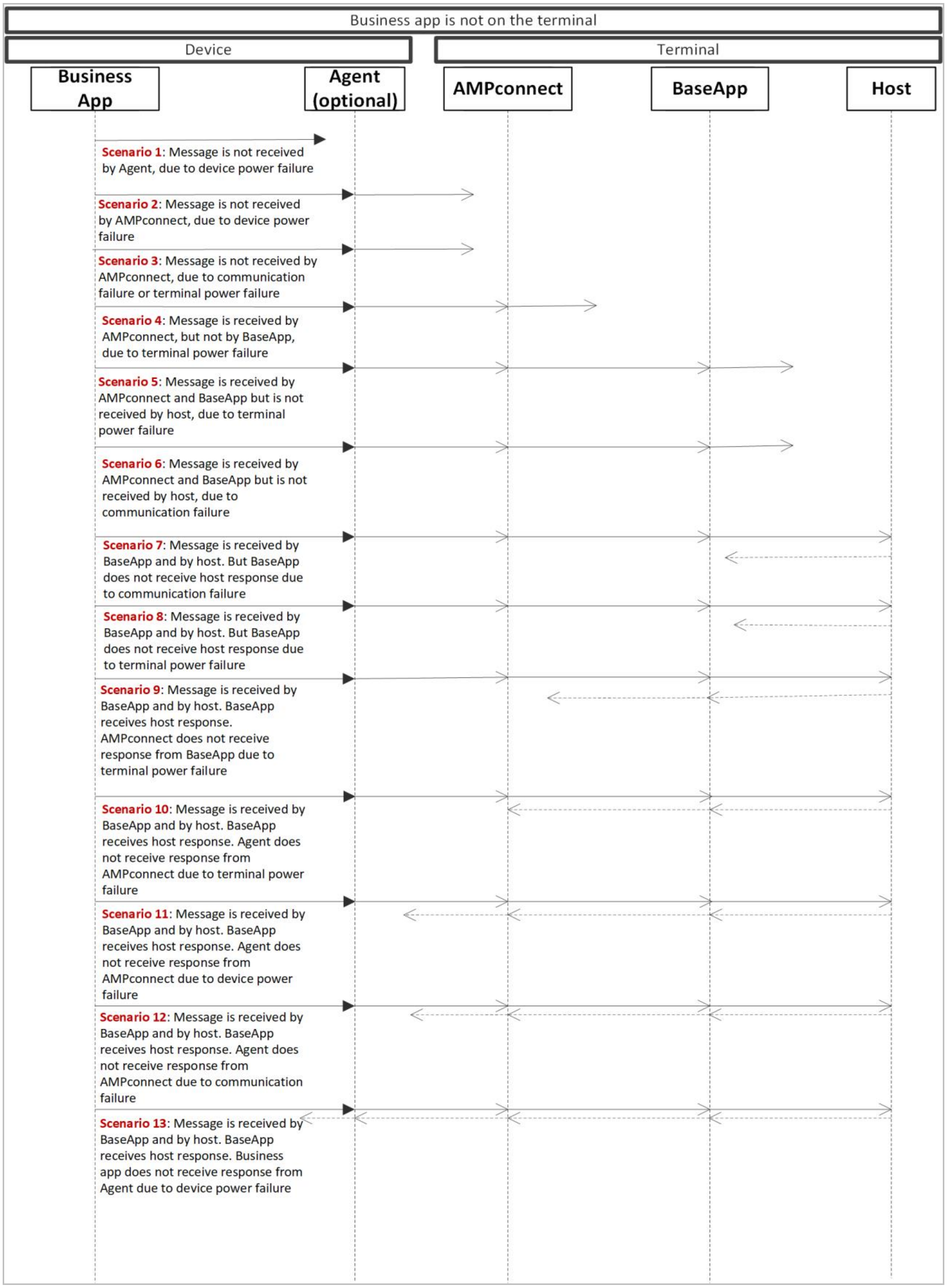

Business app is not on the terminal
Device
Terminal
Business App
Agent (optional)
AMPconnect
BaseApp
Host
Scenario 1: Message is not received by Agent, due to device power failure
Scenario 2: Message is not received by AMPconnect, due to device power failure
Scenario 3: Message is not received by AMPconnect, due to communication failure or terminal power failure
Scenario 4: Message is received by AMPconnect, but not by BaseApp, due to terminal power failure
Scenario 5: Message is received by AMPconnect and BaseApp but is not received by host, due to terminal power failure
Scenario 6: Message is received by AMPconnect and BaseApp but is not received by host, due to communication failure
Scenario 7: Message is received by BaseApp and by host. But BaseApp does not receive host response due to communication failure
Scenario 8: Message is received by BaseApp and by host. But BaseApp does not receive host response due to terminal power failure
Scenario 9: Message is received by BaseApp and by host. BaseApp receives host response. AMPconnect does not receive response from BaseApp due to terminal power failure
Scenario 10: Message is received by BaseApp and by host. BaseApp receives host response. Agent does not receive response from AMPconnect due to terminal power failure
Scenario 11: Message is received by BaseApp and by host. BaseApp receives host response. Agent does not receive response from AMPconnect due to device power failure
Scenario 12: Message is received by BaseApp and by host. BaseApp receives host response. Agent does not receive response from AMPconnect due to communication failure
Scenario 13: Message is received by BaseApp and by host. BaseApp receives host response. Business app does not receive response from Agent due to device power failure

Figure 11 - Recovery when Business App Is Not on the Terminal.

The following instructions should be followed to recover from a power or connection failure:

- Business application should always save the transaction invoice number that is included in a transaction response, in “response_invoicenum_key” tag.
- Business application should implement the following logic during start up:

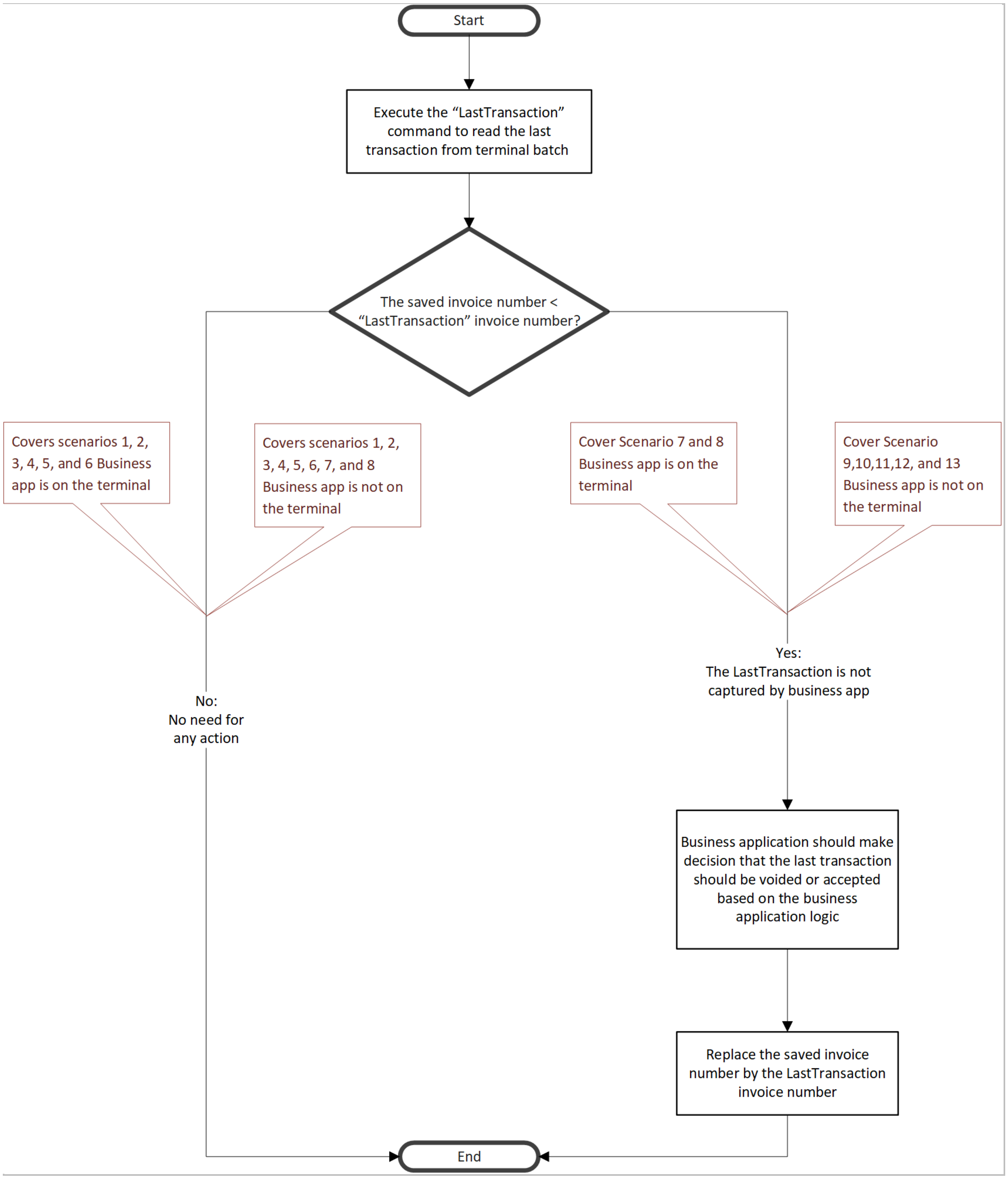
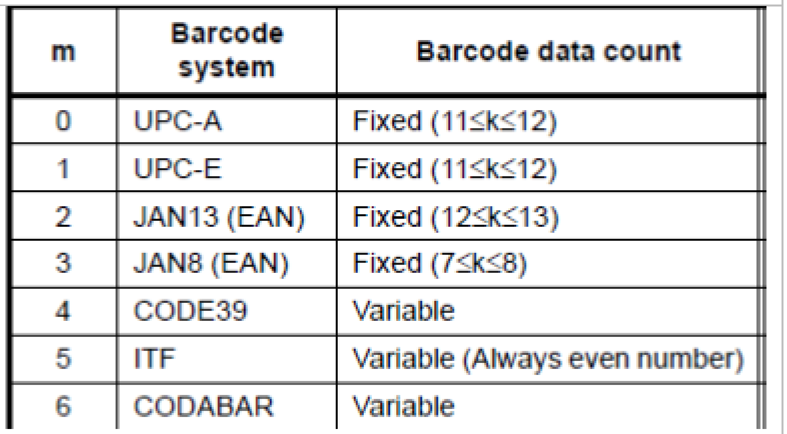

Start
Execute the "LastTransaction" command to read the last transaction from terminal batch
The saved invoice number < "LastTransaction" invoice number?
Covers scenarios 1, 2, 3, 4, 5, and 6 Business app is on the terminal
Covers scenarios 1, 2, 3, 4, 5, 6, 7, and 8 Business app is not on the terminal
Cover Scenario 7 and 8 Business app is on the terminal
Cover Scenario 9,10,11,12, and 13 Business app is not on the terminal
No: No need for any action
Yes: The LastTransaction is not captured by business app
Business application should make decision that the last transaction should be voided or accepted based on the business application logic
Replace the saved invoice number by the LastTransaction invoice number
End

Figure 12 - Recovery Scenarios.

APPENDIX F: ESCPOS COMMANDS

The following are some common ESCPOS commands that can be used during integration.

<table>
<tr><th>Command</th><th>Set command</th><th>Cancel command</th><th>Description</th></tr>
<tr><td><b>Initialization:</b>
Printer initialization
Standard mode
Line spacing
USA character select</td><td>
ESC ‘@’
ESC ‘S’
ESC ‘3’ n
ESC ‘R’ 0</td><td></td><td></td></tr>
<tr><td>Linefeed</td><td>LF</td><td></td><td></td></tr>
<tr><td>Carriage return</td><td>CR</td><td></td><td></td></tr>
<tr><td>Bold</td><td>ESC’E’1</td><td>ESC’E’0</td><td></td></tr>
<tr><td>Double strike</td><td>ESC ‘G’1</td><td>EXC’G’0</td><td></td></tr>
<tr><td>Barcode print</td><td>GS’k’m [k]</td><td></td><td>
<table>
<tr><th>m</th><th>Barcode system</th><th>Barcode data count</th></tr>
<tr><td>0</td><td>UPC-A</td><td>Fixed (11≤k≤12)</td></tr>
<tr><td>1</td><td>UPC-E</td><td>Fixed (11≤k≤12)</td></tr>
<tr><td>2</td><td>JAN13 (EAN)</td><td>Fixed (12≤k≤13)</td></tr>
<tr><td>3</td><td>JAN8 (EAN)</td><td>Fixed (7≤k≤8)</td></tr>
<tr><td>4</td><td>CODE39</td><td>Variable</td></tr>
<tr><td>5</td><td>ITF</td><td>Variable (Always even number)</td></tr>
<tr><td>6</td><td>CODABAR</td><td>Variable</td></tr>
</table></td></tr>
<tr><td>Invert print (uppside down)</td><td>ESC ‘{‘1</td><td>ESC’{‘0</td><td></td></tr>
<tr><td>Reverse print</td><td>GS’B’1</td><td>GS’B’0</td><td></td></tr>
<tr><td>Underline</td><td>ESC’_’ 1or 2</td><td>ESC’_’0</td><td></td></tr>
<tr><td>Align</td><td>ESC’a’0
ESC’a’1
ESC’a’2</td><td>ESC’a’0</td><td>Align left
Align Center
Align Right</td></tr>
<tr><td>Drawer open</td><td>EXC’p’ m n1 n2</td><td></td><td>n1: set 32h
n2: set 32h</td></tr>
<tr><td>Cutter</td><td>GS ‘V’ 0
GS ‘V’ 1</td><td></td><td>Full cut
Partial cut</td></tr>
<tr><td>Feed lines forward</td><td>ESC ‘d’ n</td><td></td><td>n: number of lines</td></tr>
</table>

Test print	DC2’t’		
Print Mode select	ESC ‘!’ n		n: font 12*24 0x00 font 8*16 0x01 Bold OR 0x08 DH OR 0x010 DW OR 0x20 UN OR 0x80

APPENDIX G: (DEPRECATED) PAYMENT

NOTE: Starting with AMP Connect version 3.0, the following endpoints are deprecated. Please use TRANSACTION endpoint.

Endpoint URL: PAYMENT.

Example, HTTP://IP_ADDRESS:22222/PAYMENT

1 PAYMENT REQUEST

The data request is in JSON format, {“cmdType”:””, “”:””, “”:””, “”:””, ...}}. There is no “ReqPayload” compare to current request format. All the parameter fields are defined in root JSON request object instead in “ReqPayload”.

2 TRANSACTION RESPONSES

To get the payment transaction result, call the “GetTransactionResult” command, which retrieves the last transaction’s data. The response data format is unified across all payment transaction commands. The following table includes the details.

JSON key	JSON key description	Value
Command	This value should be the same as the ‘cmdType” in the request.	
Status	The payment transaction’s status	“RESULT_OK” “RESULT_CANCELLED”
Seq	This value matches the data in the request.	
Payload	The actual payment transaction response data that is returned by the payment app. The actual data format is different for BaseApp and IOT Pay app as demonstrated below.	

3 DETAIL COMMANDS

3.1 PURCHASE

This endpoint calls BaseApp's Purchase / Sale transaction or the IOTPay app's Purchase transaction based on user's selection.

Purchase Request			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: calling the purchase transaction.	String	"PURCHASE"
autoPrint	Sets the external application printing status.	Boolean	TRUE – BaseApp will handle receipt printing. FALSE – the external application will handle receipt printing.
cardEntryMethod	Sets the card entry method which will be used throughout the transaction.	String	AUTO – the card is expected to be swiped, tapped, or inserted in the allocated card slot. MANUAL – the card details are expected to be entered manually.
cardNumberPAN	Sets the Primary Account Number (PAN) which will be used in the transaction.	String	String number – The card's PAN represented in string format.
expiryDate	Sets the expiry date which will be used in the transaction.	String	String date – indicates the card's expiration date in MM/YY string format.
cardSecurityCode	Sets the code for the "card not present" payment card transactions. The term for this code varies across different card brands:	String	String code – indicates the 3 or 4-digit code for the card that is to be used in the transaction. The code length depends on the card brand:

	"CSC" or "Card Security Code" – American Express. "CVC2" or "Card Validation Code" – MasterCard. "CVD" or "Card Verification Data" – Discover. "CVN2" or "Card Validation Number 2" – China UnionPay. "CVV2" or "Card Verification Value 2" – Visa. "CVC" or "Card Verification Code" – Mastercard.		AMEX – accepts 3 or 4 digits. The rest of the card brands (Visa, Mastercard, etc.) – 3 digits.
baseAmount	Sets the transaction’s base amount.	String	String amt – an 8-digit numeric value in string format that represents the transaction’s base amount. The general format for this value is “999999.99”. If presented differently, BaseApp modifies the value. For example: 1.23 is reflected as 1.23. 123 is reflected as 123.00. 12.3 is reflected as 12.30. 123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.
tipAmount	Sets the transaction’s tip amount.	String	String amt – an 8-digit numeric value in string format that represents the transaction’s tip amount. The general format of this value is “999999.99”. If presented differently, BaseApp modifies the value. For example: 1.23 is reflected as 1.23. 123 is reflected as123.00. 12.3 is reflected as 12.30.

			123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.
address	Sets the card holder's address.	String	String address: the card holder's address. Max length: 256 characters.
zipCode	Sets the card holder's zip code.	String	String zip: the card holder's zip code. Max length: 256 characters.
echoData	Sets a unique identifier to be used by the external application for its internal processes.	String	String brn – a 50-byte character that represents a unique identifier for the transaction.
cashbackAmount	Sets the transaction's cashback amount.	String	String cashback – an 8-digit numeric value in string format that represents the transaction's cashback amount. The general format for this value is "999999.99". If presented differently, BaseApp modifies the value. For example: 1.23 is reflected as 1.23. 123 is reflected as 123.00. 12.3 is reflected as 12.30. 123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.
taxAmount	Sets the transaction's tax amount.	String	String tax – an 8-digit numeric value in string format that represents the transaction's tax amount. The general format for this value is "999999.99". If presented differently, BaseApp modifies the value. For example: 1.23 is reflected as 1.23.

			123 is reflected as 123.00. 12.3 is reflected as 12.30. 123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.

Example:

Request to AMP Connect:

```
http://10.242.3.90:22222/PAYMENT
{"baseAmount":"5.99","cardEntryMethod":"AUTO","echoData":"123456789"
,"autoPrint":”true”,"cmdType":"PURCHASE"}
```

Response (The communication status response from AMP Connect):

```
{“AMPConnectResponseCode”:0}
```

3.2 REFUND

This endpoint calls BaseApp’s refund function. (the IOTPay app doesn’t yet support refund)

Refund			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: requesting refund.	String	“REFUND”
autoPrint	Sets the external application printing status.	Boolean	TRUE – BaseApp will handle receipt printing. FALSE – the external application will handle receipt printing.
baseAmount	Sets the transaction’s base amount.	String	String amt – an 8-digit numeric value in string format that represents the transaction’s base amount. The general format for this value is “999999.99”. If presented differently,

			BaseApp modifies the value. For example: 1.23 is reflected as 1.23. 123 is reflected as 123.00. 12.3 is reflected as 12.30. 123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.
cardEntryMethod	Sets the card entry method which will be used throughout the transaction.	String	AUTO – the card is expected to be swiped, tapped, or inserted in the allocated card slot. MANUAL – the card details are expected to be entered manually.
echoData	Sets a unique identifier to be used by the external application for its internal processes.	String	String brn – a 50-byte character that represents a unique identifier for the transaction.

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{“autoPrint”:”TRUE”,"baseAmount":"14","cardEntryMethod":"AUTO","echo
Data":"123456789","cmdType":"REFUND"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.3 VOID

This endpoint calls BaseApp’s void function.

VOID			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: requesting void.	String	“VOID”

voidType	The type of void transaction.	String	INVOICE_NUMBER TRACE_NUMBER RRN_NUMBER AUTH_NUMBER CARD_NUMBER
voidNumber	The number of the void transaction	Boolean	String number: e.g. 0000018
autoPrint	Sets the external application printing status.	Boolean	TRUE – BaseApp will handle receipt printing. FALSE – the external application will handle receipt printing.
echoData	Sets a unique identifier to be used by the external application for its internal processes.	String	String brn – a 50-byte character that represents a unique identifier for the transaction.

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{“autoPrint”:”TRUE”,"echoData":"12345678910","voidNumber":"0000019",
"voidType":"INVOICE_NUMBER","cmdType":"VOID"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.4 PRE-AUTH

This endpoint calls BaseApp’s pre-auth function.

Pre-Authorization			
JSON key	**JSON key description**	**Value type**	**Value**

cmdType	Command type or the intended action: request for preauthorization.	String	“PRE_AUTHORIZATION”
autoPrint	Sets the external application printing status.	Boolean	TRUE – BaseApp will handle receipt printing. FALSE – the external application will handle receipt printing.
baseAmount	Sets the transaction’s base amount.	String	String amt – an 8-digit numeric value in string format that represents the transaction’s base amount. The general format for this value is “999999.99”. If presented differently, BaseApp modifies the value. For example: 1.23 is reflected as 1.23. 123 is reflected as 123.00. 12.3 is reflected as 12.30. 123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.
cardEntryMethod	Sets the card entry method which will be used throughout the transaction.	String	AUTO – the card is expected to be swiped, tapped, or inserted in the allocated card slot. MANUAL – the card details are expected to be entered manually.
echoData	Sets a unique identifier to be used by the external application for its internal processes.	String	String brn – a 50-byte character that represents a unique identifier for the transaction.

Example

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"autoPrint":"TRUE","baseAmount":"13","cardEntryMethod":"AUTO","echo
Data":"123456789","cmdType":"PRE_AUTHORIZATION"}
```

- Response:

```
{"AMPConnectResponseCode":0}
```

3.5 COMPLETION

This endpoint calls BaseApp's completion function.

Completion			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: request for Completion	String	"COMPLETION"
completionType	The type of complete transaction.	String	LAST_TRANSACTION TRACE_NUMBER INVOICE_NUMBER REFERENCE_NUMBER BROWSE_ALL
completionNumber	The number of the void transaction	Boolean	String number: e.g. 13
autoPrint	Sets the external application printing status.	Boolean	TRUE – BaseApp will handle receipt printing. FALSE – the external application will handle receipt printing.
echoData	Sets a unique identifier to be used by the external application for its internal processes.	String	String brn – a 50-byte character that represents a unique identifier for the transaction.

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{“autoPrint”:”TRUE”,"completionType":"INVOICE_NUMBER","echoData":"12
3456789","compleNumber":"13","cmdType":"COMPLETION"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.6 FORCE POST

This endpoint calls BaseApp’s force post function.

Force Post			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	The command type or the intended action: request for force post.	String	“FORCE_POST”
autoPrint	Sets the external application printing status.	Boolean	TRUE – BaseApp will handle receipt printing. FALSE – the external application will handle receipt printing.
baseAmount	Sets the transaction’s base amount.	String	String amt – an 8-digit numeric value in string format that represents the transaction’s base amount. The general format for this value is “999999.99”. If presented differently, BaseApp modifies the value. For example: 1.23 is reflected as 1.23. 123 is reflected as 123.00. 12.3 is reflected as 12.30. 123. is reflected as 123.00. 1.234 is reflected as 1.23. 1.235 is reflected as 1.24.

cardEntryMethod	Sets the card entry method which will be used throughout the transaction.	String	AUTO – the card is expected to be swiped, tapped, or inserted in the allocated card slot. MANUAL – the card details are expected to be entered manually.
echoData	Sets a unique identifier to be used by the external application for its internal processes.	String	String brn – a 50-byte character that represents a unique identifier for the transaction.

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{“autoPrint”:”TRUE”,"baseAmount":"13","cardEntryMethod":"AUTO","echo
Data":"123456789","cmdType":"FORCE_POST"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.7 SETTLEMENT

This endpoint calls BaseApp’s settlement function.

Settlement			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: requesting settlement.	String	“SETTLEMENT”

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"cmdType":"SETTLEMENT"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.8 FORCE BATCH CLOSE

This endpoint calls BaseApp’s force batch close function.

Force Batch Close			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: requesting force batch close.	String	“FORCE_BATCH_CLOSE”

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"cmdType":" FORCE_BATCH_CLOSE"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.9 KEY EXCHANGE

This endpoint calls BaseApp’s key exchange function.

Key Exchange			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: requesting key exchange.	String	“KEY_EXCHANGE”

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"cmdType":" KEY_EXCHANGE"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.10 PARAMETER DOWNLOAD

This endpoint calls BaseApp’s parameter download function.

Parameter Download			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: requesting Parameter Download.	String	“DL_PARAM”

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"cmdType":" DL_PARAM"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.11 EMV DOWNLOAD

This endpoint calls BaseApp’s EMV download function.

EMV Download			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: requesting EMV Download.	String	“DL_EMV”

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"cmdType":"DL_EMV"}
```

- Response:

```
{“AMPConnectResponseCode”:0}
```

3.12 TERMINAL INITIALIZATION

This endpoint calls BaseApp’s terminal initialization function.

Terminal Initialization			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: requesting Terminal initialization.	String	“TERMINAL_INITIALIZATION”

Example:

- Request:

```
http://10.242.3.36:22222/PAYMENT
{"cmdType":" TERMINAL_INITIALIZATION"}
```

- Response:

```
 {“AMPConnectResponseCode”:0}
```

3.13 GETTING TRANSACTION RESULTS

The third-party application should regularly call this function (right after the payment transaction’s API call) in order to get the transaction results.

Getting Transaction Results			
JSON key	**JSON key description**	**Value type**	**Value**
cmdType	Command type or the intended action: request for transaction results.	String	“GET_TRANSACTION_RESULT”

Example:

- Request:

```
http://192.168.0.108:22222/PAYMENT
{"cmdType":"GET_TRANSACTION_RESULT"}
```