

Terminal Test Scenarios

- Introduction
- Prerequisites
- Document On-boarding Process
- Transactions
 - Create
 - Steps to Test:
 - Acceptance Criteria
 - Cancel (Terminal and ISV software initiated)
 - Steps to Test
 - Acceptance Criteria
 - Timeout
 - Steps to Test
 - Acceptance Criteria
 - Terminal busy
 - Steps to Test
 - Acceptance Criteria
 - Save
 - Steps to Test
 - Acceptance Criteria
 - Non-Terminal Transaction Test Scenarios
 - Index (Transaction Recovery)
 - Steps to Test
 - Acceptance Criteria

Introduction

The purpose of this document is to provide steps to test and acceptance criteria to <Partner>; i.e. communicate our expectations prior to certification testing and validation.

NOTE: all tests that require a terminal will be performed on a PAX device.

Prerequisites

- PF provisioning:
 - PFG test account
 - Loopback processor
 - Control Center (use the partner sandbox organization)
 - Organization
 - Merchant
 - Source
 - Terminal
- UAT environment
 - Ability to onboard a test partner client
 - A new merchant; two merchants if supporting multi-MID support is required
 - A new terminal
 - Ability to run transaction(s)

- Others(?)

Document On-boarding Process

This section attempts to capture information that will be useful for Bank Of America healthcare payment solutions implementation team when provisioning productions accounts in the future. Capture(provided by Partner) the steps needed for gathering the individual payment to be sent to the processor.

<TODO: get provisioning steps from the Partner>

i.e. capture steps required by the partner to provision an account in their software.

Transactions

Create

Trigger amount:

- PFG: (Approval using \$1.00, Decline using \$0.12)

Steps to Test:

1. Initiate transaction on the terminal.
2. The terminal will prompt for card holder data, Input test card.
3. Verify transaction result in integrated application.

Acceptance Criteria

1. As an integrated application I can initiate a transaction using **serial_number** or **client_id** and receive a transaction broker id (**tb_id**).
2. Transaction results:
 - a. As an integrated application I can receive the transaction details via a callback url and receive a transaction result that includes the **tb_id** from step 1.
 - b. As an integrated application I can verify that I received the Remote ID.
 - c. The partner's software displayed a receipt.

Reference: [Positive Terminal Transaction \(Callback\)](#)

tb_id	comment(s)/screenshot
1 <tb_id>	EMV Approval
1 <tb_id>	EMV Decline
1 <tb_id>	Keyed Approval
1 <tb_id>	Keyed Decline
1 <tb_id>	MSR Approval
1 <tb_id>	MSR Decline
1 <tb_id>	Contactless Approval
1 <tb_id>	Contactless Decline

NOTE: screen shot rendered receipt for the receipt section below.

Cancel (Terminal and ISV software initiated)

Steps to Test

1. Initiate transaction to terminal.
2. Initiate cancel request via terminal.

Acceptance Criteria

1. As an integrated application I can initiate a transaction using serial_number and receive a transaction broker id (**tb_id**).
2. Transaction results:
 - a. As an integrated application I can receive the transaction details via a callback url and receive a transaction result of canceled.

Reference: [User Cancel Terminal Transaction \(Callback\)](#)

tb_id	comment(s)/screenshot
1 <tb_id>	Terminal initiated
1 <tb_id>	ISV initiated

Timeout

Steps to Test

1. Initiate transaction to terminal.
2. Wait for the transaction to timeout.

Acceptance Criteria

1. As an integrated application I can initiate a transaction using serial_number and receive a transaction broker id (**tb_id**).
2. Transaction results:
 - a. As an integrated application I can receive the transaction details via a callback url and receive a transaction result of timeout.

Reference: [Timeout Terminal Transaction \(Callback\)](#)

tb_id	comment(s)/screenshot
1 <tb_id>	

Terminal busy

Steps to Test

1. Initiate a terminal transaction.
2. Initiate another terminal transaction.

Acceptance Criteria

1. As an integrated application I can receive a response indicating the terminal is busy.

Reference: TBD

tb_id	comment(s)/screenshot
1 <tb_id>	

Save

Steps to Test

1. Initiate transaction to terminal
2. Input test card
3. Store token and account information in integrated application

Acceptance Criteria

1. As an integrated application I can receive the payment token via a [callback url|polling]

tb_id	comment(s)/screenshot
1 <tb_id>	EMV
1 <tb_id>	Keyed
1 <tb_id>	MSR
1 <tb_id>	Contactless

Non-Terminal Transaction Test Scenarios

Reference this [page](#) to test these cases.

Index (Transaction Recovery)

Steps to Test

1. Initiate a transaction with out a callback url.
2. Input test card
3. Verify transaction result in integrated application

Acceptance Criteria

1. As an integrated application I can initiate a index call to view the transaction details for the transaction that did not perform a callback.

Reference: TBD

tb_id	comment(s)/screenshot
N/A	