

Description: logo.wmf

Automation Solution for testing of Peripheral devices.

**Date: September 24th, 2018**

1. **Objective**

Advance America is automating the Qfundx application using UI automation framework. During the flow of services, the Qfundx application will interact with multiple external peripheral devices. As availability of the peripheral devices is a constraint, there is need to provide a solution to provision the test environments with substitute of peripheral devices.

The benefits of the solutions are to have seamless execution of UI automation scripts without having dependency on the peripheral systems.

1. **List of devices**

Following are the key items to be considered for testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Peripheral device** | **Service Based/Not-service based** | **Possible solution would involve** | **In Scope/Out of Scope** |
| **Scanner** | Not service-based hardware device | A stub which mimics this application-scanner interaction | In Scope |
| **Debit Card Reader** | Service Based peripheral device | A service mockup of this interaction | In Scope |
| **MICR Reader** | Not sure | This field is editable in the application so does not require peripheral automation | Out of scope |
| **Signature Pad** | I will get you this info | I will get you this info | Pending with AA for further inputs |

1. **Mode of Interaction**

The Qfundx systems will interact with the peripheral systems in two different methods.

1. Service based interactions - The interactions are based over HTTP / HTTPS protocols and data flows between the systems using Xml or JSON.
2. Non-Service Based interactions - The interactions are based on Java / .Net objects.
3. **Solutions approach**
4. Service based interactions – When the integrations are service based. A due diligence is carried out to understand the kind of integration by knowing the data and transport protocol used over there. Based on the data we use multiple open source tools like mock server or wire mock to simulate the backend systems. There will be an intelligent mechanism with in the mocks to determine if real device is available or not, based the status the call is routed between real system and mock. We will open source framework / tools like Mock server and Wirework to implement the same.

Peripheral devices if exists

Mocks are created based on RR pairs

IF Real device

Is not available

Peripheral devices if does not exists (MOCK)

1. Non-Service Based interactions - When the interactions are not service based, we assume the interactions will be happen via direct objects to the embedded systems. We need to understand how the data captured from the device is passed on to the applications and how application acknowledges. There are multiple stages and applications in between the UI application and the device. We need to identify the right layer where the transaction data can be captured and build logic to respond back. There are tools like Junit, which will help us to do this activity.

**Before mocking**

Peripheral devices if exists

**After mocking**

Peripheral devices if does not exists (MOCK)

Peripheral devices if exists

IF Real device

Is not available

Mocks are created based on java objects