

Supported Devices

IPS Enterprise and IPS Encrypted Card Readers

The IPS devices are audio jack-connected card readers. Unlike the IDTECH Shuttle, these readers are powered by an internal battery. The IPS devices have a fast startup time and do not produce a constant tone through the audio jack.

Because the IPS devices connect through the audio port and there is no way to immediately detect the device type, you will receive a `deviceConnected:` event even if the user has only plugged in headphones. Since there is no activation with the IPS devices, a `deviceActivated:` and `deviceBecameReadyForSwipe:` will also be sent immediately. In order to be sure that the device is an IPS Enterprise or IPS, the `PGSwipeHelpsEnterprise` and `PGSwipeHelps` provide a `beginTestCommunication:` method you can use to attempt to communicate with the devices. Success is returned when the device has been successfully identified. This is not done by default to eliminate a delay before the device becomes active.

iDynamo

The iDynamo connects to the mobile device via Apple's dock connector and is only compatible with iOS devices that use the older 30-pin (non-Lightning) dock connector.

When physically attached, the iDynamo is almost immediately ready to receive swipe events. When connected, the Swipe Delegate should expect a `deviceConnected:` message, immediately followed by a `deviceActivationFinished:` message, then a `deviceBecameReadyForSwipe:` message.

When the device is physically detached, the delegate receives the events in reverse order, i.e. `deviceBecameUnreadyForSwipe:`, `deviceDeactivated:`, `deviceDisconnected:`.

App Store: To support the iDynamo on an app distributed through the App Store, Apple may require you to contact MagTek for information before they will process your submission. To disable iDynamo support, do not add it to "Supported external accessory protocols" in your `info.plist`. You will still receive connect and disconnect events, but activation will fail, so be sure to check if the sending device is the iDynamo object and ignore it if so.

Known Issue with the iDynamo: There is an issue with device disconnection with the iDynamo and iOS's ExternalAccessory framework. Upon disconnection, the stream communicating with the device is closed, during which you may receive the warning: `[NSCondition dealloc]: condition (<NSCondition: 0x1d54ce90> '(null)') deallocated while still in use`. After reconnecting, a later disconnect may randomly cause the app to crash with an attempt to send a message to the deallocated instance. This does not occur frequently, and is more likely to occur when rapidly opening and closing the application (which sends a disconnect followed by a reconnect when the app re-opens). This issue is with Apple's accessory-handling framework. Apple is aware of the issue and may fix it in a future iOS release.

IDTECH Shuttle

The Shuttle (referred to in code as a UniMag device) is an audio jack-connected card reader. It is powered by a tone from the iPod / iPad / mobile phone. Before the Shuttle can receive swipes, it must be powered up.

Because the Shuttle connects through the audio port and there is no way to detect the device type until the device is activated, you will receive `deviceConnected` events whenever any device is attached to that port. For example, if the user attaches headphones, you will receive a connection event from the Mobile SDK.

The Mobile SDK can be configured to automatically attempt to power on the swipe reader immediately (this is the default), or you can disable the automatic activation and only activate the device when desired (e.g. on a payment screen, or when the user clicks a button).

Important: When powering on the device, the audio volume must be at maximum (done automatically by default). The tone generated through the audio port to activate the device can be very painful to a listener if they have connected speakers or headphones. For this reason, `swipeController.uniMagReader.messageOptions.activateReaderWithoutPromptingUser` is set to NO by default, causing the SDK to prompt the user for confirmation before activating the reader.

The Shuttle saves battery by only allowing swipes when a swipe has been requested, and a timeout occurs if a swipe is not received quickly enough (20 seconds by default). For simplicity, the SDK defaults to automatically requesting a swipe on activation and continuously renewing the swipe request. If you have issues with battery life, you can set `swipeController.uniMagReader.alwaysAcceptSwipe` to NO and manually call `[swipeController.uniMagReader requestSwipe]` when ready for a swipe.