Universal SDK Demo App
QuickStart Guide

11 July 2023
Rev. H

ID TECH
10721 Walker Street
Cypress, CA 90630 USA

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

Revision History

Date	Rev	Changes	By
07/11/2023	H	Reimplemented change history. Updated various requirements/supported products. Updated links throughout document.	CB

Table of Contents

1. Overview

This document provides a detailed description of and instructions for using the ID TECH Universal SDK Demo app.

ID TECH provides the Universal SDK, a Software Development Kit, to customers to develop payment applications around various ID TECH products. The Universal SDK offers developers a unified API for communicating with ID TECH devices and enables the rapid development of applications written in a Visual Studio (.Net 4.8, .Net Standard 2.1, .Net 6.0, .Net 7.0). The API library is available as a NuGet module for integrating directly into a Visual Studio Project.

Using the Universal SDK, a developer can work with built-in connectivity modules, avoiding any need to write USB or serial I/O routines, and take advantage of convenience classes that aid in data-parsing, event handling, transaction flow, error detection, and so forth. Because the same APIs are used across multiple products, code written for one ID TECH device can often be used with other ID TECH devices, greatly reducing the time spent writing one-off code to support individual devices. Solutions can produced in Visual Studio for Windows, Mac and Linux.

The Universal SDK comes with sample code showing how to use the SDK to conduct MSR, contact EMV, and contactless transactions. One of the apps included is the USDK Demo app, also available as a download on the Visual Studio project page.

The USDK Demo app offers a quick, easy way to establish communication with an ID TECH reader, query its firmware version (and key status), send low-level commands, initiate transactions, and so on. Using the USDK Demo app's graphical user interface, users can quickly run various Universal SDK commands in a point-and-click manner, then see detailed log messages to understand the kinds of messages that are sent back and forth from the card reader.

Familiarity with the USDK Demo app aids not only learning how to work with ID TECH hardware but also in dealing with various types of support issues. For example, for issues requiring technical support, ID TECH's Tech Support department may ask a user to run the **Get Firmware** command (or other commands) to aid the troubleshooting process.

ID TECH recommends that all customers become familiar with the USDK Demo app. This document will get users up to speed quickly in using this important utility.

2. How to Obtain the USDK Demo App

The latest version of the USDK Demo app is always available on the ID TECH Visual Studio project page. The app is free to download and does not require users to download the entire SDK. No registration is required. It is supported only on Windows.

To obtain the a version of the Universal SDK for another platform (iOS, Android, GCC), check the Product page for the product in question, as listed on the Knowledge Base.

3. Applicable Products

The Universal SDK Demo app currently supports the following devices, using USB connectivity:

- UniPay
- Vendi
- Vendi III
- Kiosk III and IV
- MiniSmart II
- Spectrum Pro
- L100
- CM100
- Augusta
- Augusta KB
- UniPay 1.5
- VP3300
- VP8800
- SecureMag
- K100
- TMS
- SecureKey
- SREDKey 2
- NEO2 devices
- NEO3 devices

The Universal SDK Demo app also supports RS-232 communication with the following products:

- Vendi
- Vendi III
- MiniSmart II
- Spectrum Pro
- Kiosk III and IV
- L100
- BT Mag
- VP8800
- SecureMag
- K100
- NEO2 devices
- NEO3 devices

The Universal SDK Demo app also supports IP communication with the following products:

- NEO2 devices: VP6800, VP5300
- NEO3 devices: VP7200, Kiosk V

These lists can and will change as more products become supported. Refer to the download URL given above for the latest version of the app.

This document assumes users are running version 3.2.4.383 of the USDK Demo app (or later).

4. Installation

The Universal SDK Demo app comes with a Window installer.

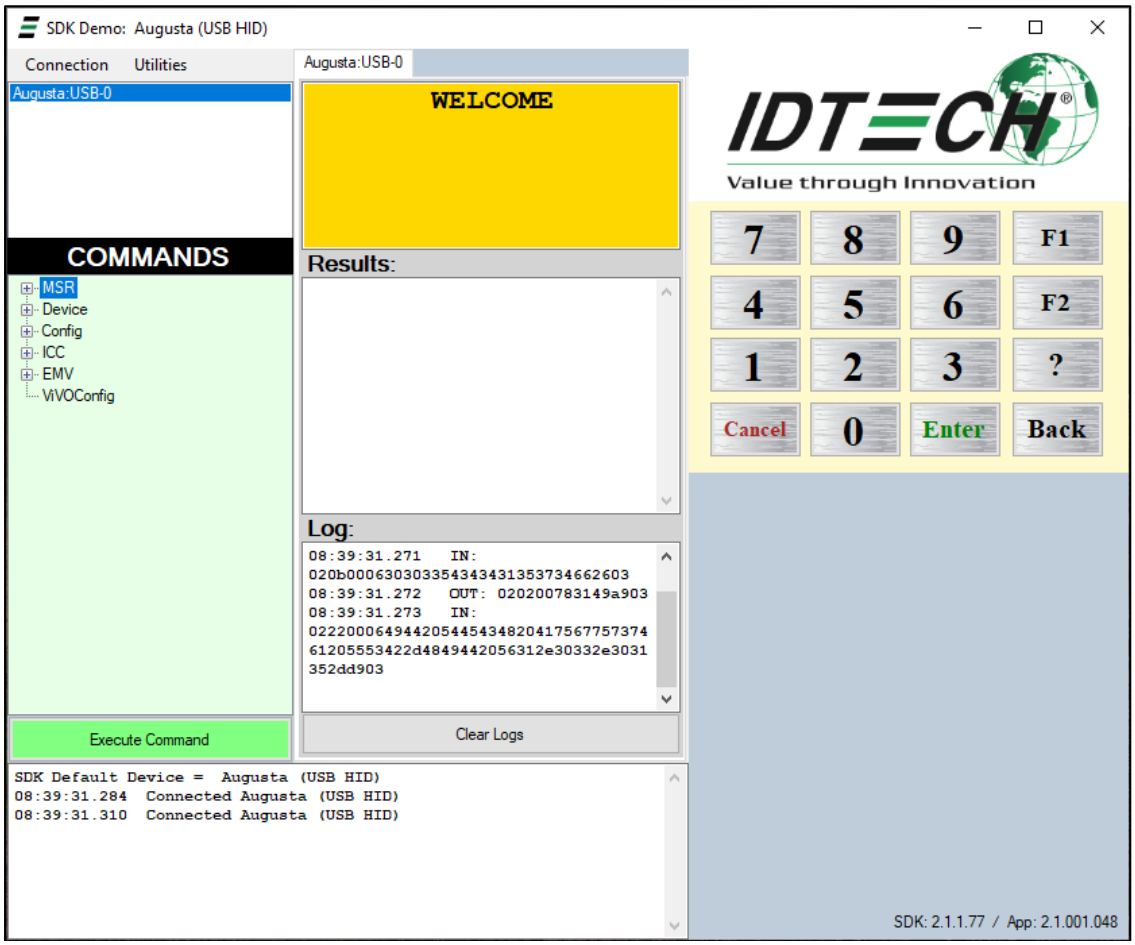
4.1. Installation Requirements

- Windows 7 or later (the Universal SDK Demo app requires a minimum version of Microsoft .NET 4.8 or later, which is only supported from Windows 7 on)
- 13 megabytes of disk space

Note: Uninstall any previous versions of the Universal SDK Demo app before installing a newer version.

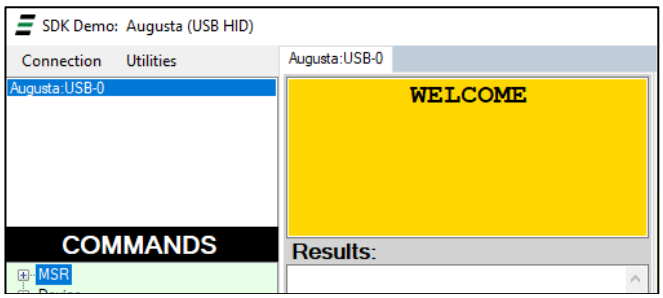
5. Launching the Program

If a shortcut is not available, double-click the **UniversalSDKDemo.exe** file in the **ID Tech** folder where the application was installed.



6. Connecting a Device

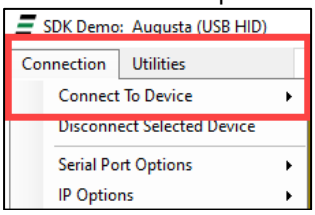
The Universal SDK Demo app supports hot-swapping and will auto-detect any compatible device a few seconds after the device is plugged in. Allow several seconds for the device to appear in the connection panel of the main UI window.



Note: When connecting to a USB device for the first time, Windows may pause to search for a device driver. This may cause a delay of up to a minute before the device registers as "connected" on the UI. In most instances, ID TECH devices do not require special drivers. No user action of any kind is needed; simply wait for the device to appear.

Serial (RS-232) devices will attempt to connect to the devices default baud rate and will attempt to connect to the available port. If more than one port is available, it will prompt you to pick one. If you require a different **Baud Rate**, to change the Baud Rate:

1. Click Connection.
2. Click Serial Port Options.



3. Select the desired setting to change.

When a device connects, the **Results** panel displays a result in the following format with the name of the connected device:

```
SDK Default Device = Augusta (USB HID)
08:39:31.284 Connected Augusta (USB HID
```

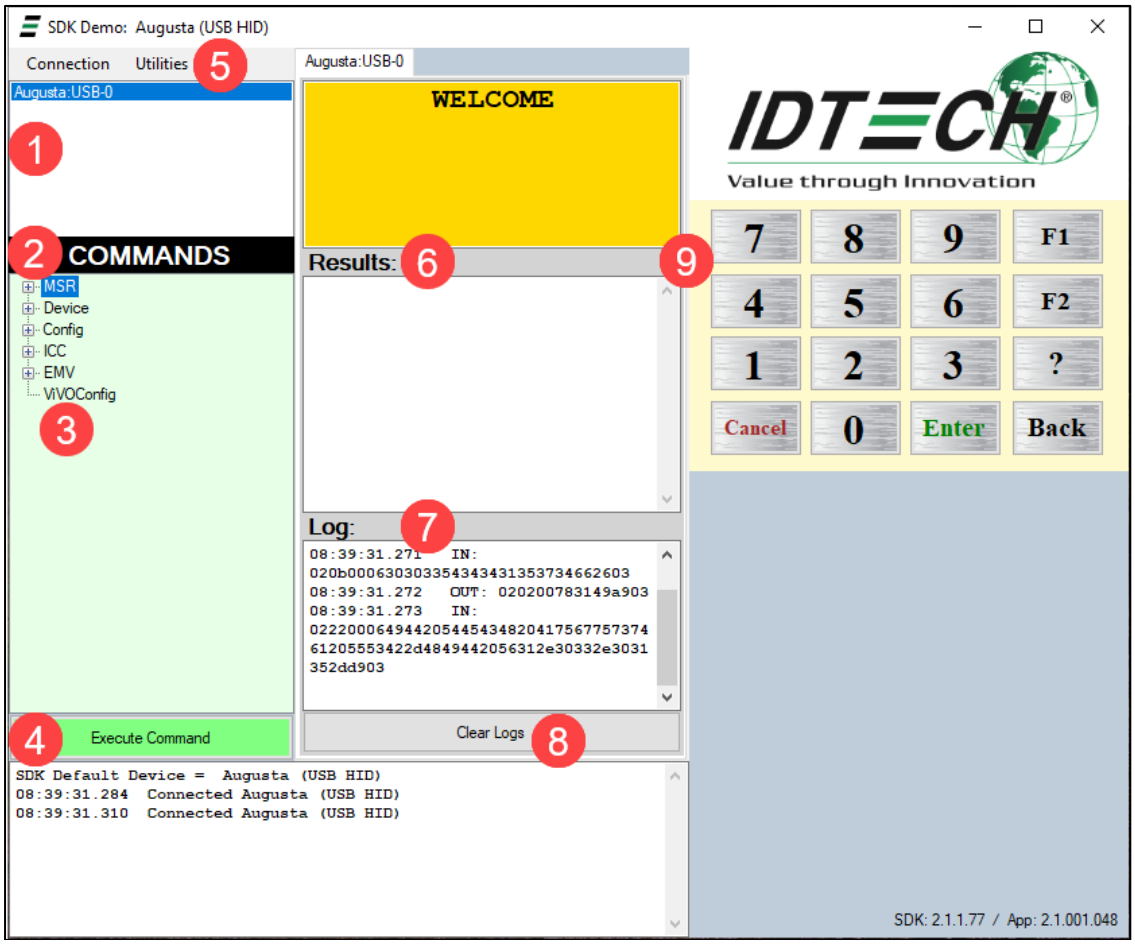
6.1. Troubleshooting Device Connections

If the Universal SDK Demo app does not connect to the device:

1. Unplug it and plug it back in.
2. Re-launch the Universal SDK Demo app.
3. Check the **Windows Device Manager** (or **Devices and Printers**) to make sure the device is listed.
4. If the device requires power, make sure the appropriate power supply is connected.

7. Main Window Overview

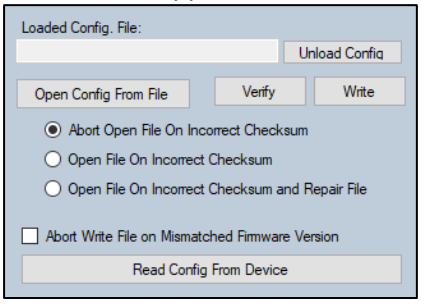
Refer to the screenshot below.



7.1. Main Sections

The Universal SDK Demo app window is divided into several main sections.

1. The **Connection** panel and menu (in the upper left corner) shows connection info, including the type of device to which the program is connected. The **Connection** menu provides options for connecting specific device types, such as RS-232 devices.
2. The **Commands** panel contains a tree of available commands. The tree-based lists in this control update dynamically according to the type of device connected; not all commands are the same for all devices. For example, the **CTLS** tree node is available only for devices that support Contactless transactions. It will not appear for other connected devices.
3. ViVOconfig: Clicking ViVOconfig in the Command tree displays a configuration menu in the USDK Demo app's context-sensitive area:

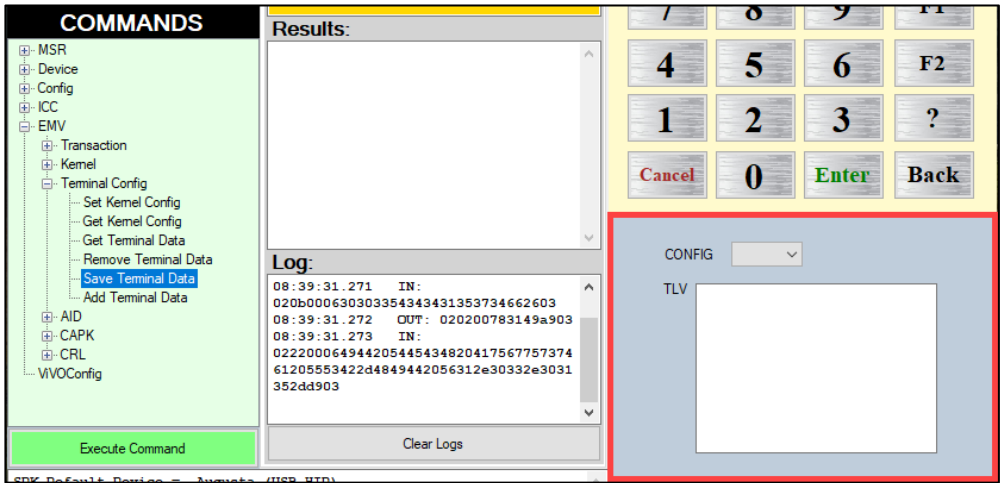


ViVOconfig features include reading and saving configurations from a device, saving those configurations, and writing configurations to devices. For a full guide to ViVOconfig, see the ViVOconfig User Manual on the ID TECH Knowledge Base.

4. Execute any command listed in the command tree by first highlighting (single-clicking) the command name, then clicking the **Execute Command** button (or double-click a command name in the tree to execute it).
5. The **Utilities** menu provides two features:
    a. **Decryption:** when clicked, this button displays a dialog in which users can insert data values that will allow the Universal SDK Demo app to derive DUKPT keys and decrypt transaction data.
    b. **Parsomatic:** when clicked, this button opens the ID TECH Parsomatic in a web browser; use this tool to parse ID TECH card-reader data streams from AR, GR, NGA, NEO I, and NEO II devices.
6. The **Results** panel displays notifications that go beyond the raw message data of the Log panel such as status messages, error codes, track data, and TLVs.
7. The **Log** panel shows log messages representing message traffic going from the Universal SDK Demo app to the card reader (OUT), and messages from the card reader demo app (IN).
8. Click the **Clear Logs** button at any time to erase Log and Results messages.
9. The PIN Pad emulator simulates PIN pad functionality and includes an LCD window for showing EMV "LCD messages".

8. Context-Sensitive Area

The area immediately to the right of the **Log** panel is context sensitive. When selected commands have no extra fields, it remains grayed out or hidden. When relevant, it displays text areas and control elements for commands. For example, when the viewing the list of EMV-related commands, selecting the **Get Terminal Data** command shows extra controls labeled **CONFIG** and **TLV**:



In general, any context-sensitive controls that appear in this special area change (or specify) parameters to the command in question.

9. Common Commands

The following commands are commonly used with ViVOtech products and are in the USDK Demo app's command tree. See the Universal SDK documentation or *Interface Developer's Guide* for each product (NEO, NEO 2, Augusta, and AR) for complete details about the commands below and their implementation.

9.1. Get Serial Number

The **Get Serial Number** command resides under the **Config** node of the command tree. Run this command to see the device's serial number in the **Results** panel.

9.2. Get Firmware Version

ID TECH Support often asks users for their device's firmware version. Obtain this by running the Firmware Version command, located under the **Device** node of the command tree. Click the **+** symbol to open the **Device** node, then double-click the **Firmware Version** command. Check the message in the **Results** panel to see the firmware version.

9.3. Update Device Firmware

The **Update Device Firmware** command initiates a reader's firmware update process. Updating firmware requires the appropriate firmware file, available from your ID TECH representative.

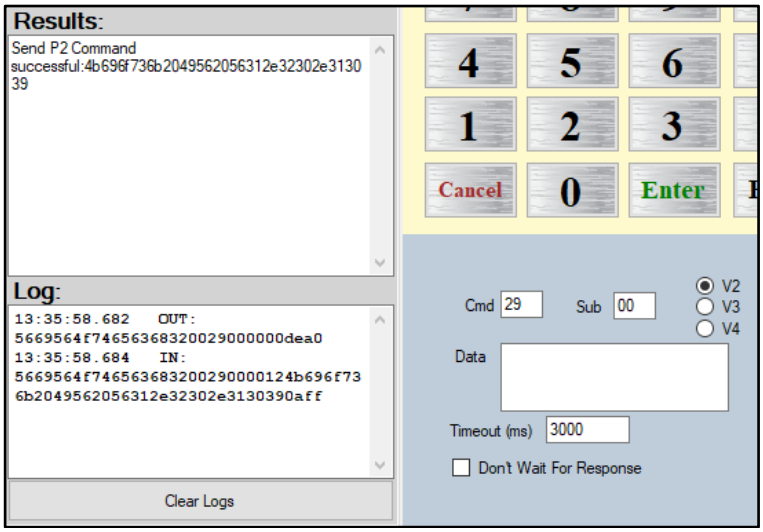
Because the USDK Demo app updates several types of firmware, the **Update Device Firmware** command is context-sensitive and changes depending on the product connected to the computer. Most ID TECH devices have a single processor and thus a single **Update Device Firmware** option; the only ID TECH device with more than one processor is the VP6800, which has options for each processor—an **Update Device K81 Firmware** command and an **Update 1050/K81/Bootloader** command. Note that each processor has a separate firmware file, as does the bootloader.

Start the firmware update by clicking the appropriate command under the command tree's **Devices** node. The USDK Demo app prompts the user to select a firmware or bootloader file in a File Explorer window. After the user selects that file, the **Results** panel shows the device restarting, entering the bootloader, updating the firmware, and restarting again.

9.4. Send NEO Command

Send NEO Command is a context-sensitive command listed under the **Device** node. This command provides a fast way to test any firmware command in the Universal SDK Demo app environment without writing or compiling code.

The **Send Data Command** sends any supported NEO firmware command to the target device (find available commands in the *Interface Developer's Guide* for each device). Users can send commands to the device not yet exposed in the USDK high-level API (or that are exposed but not implemented in the Universal SDK Demo app GUI).



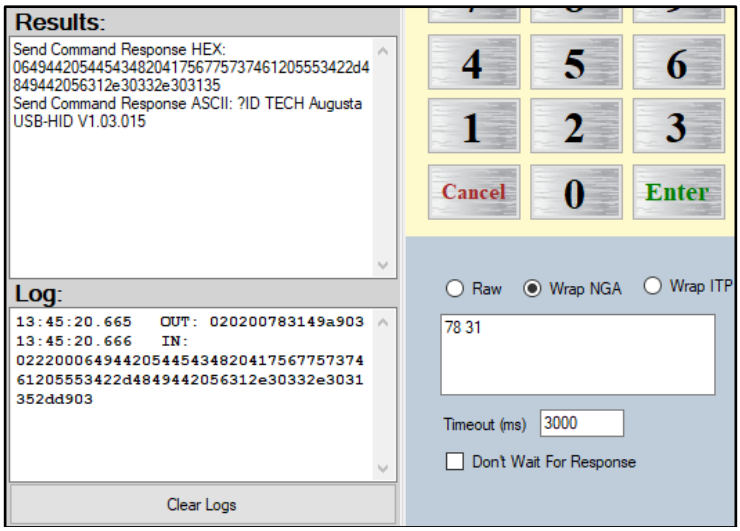
In the screen shot above:

- The connected device is a SREDKey 2 device sending the **Get Extended Firmware Version** command, which is **29 00** for Kiosk III and IV units.
- The user has entered command **29** in the command field and 00 as the sub-command.
- The **Results** panel displays the current firmware version in hexadecimal: **4b696f736b2049562056312e32302e313039**, which converts into ASCII as "Kiosk IV V1.20.109."

9.5. Send Data Command

The **Send Data Command** is a context-sensitive command listed under the **Device** node. This command provides a fast way to test any firmware command in the Universal SDK Demo app environment without writing or compiling code.

The **Send Data Command** sends any supported raw firmware command to the target device (find available commands in the *Interface Developer's Guide* for each device). Users can send commands to the device not yet exposed in the USDK high-level API (or that are exposed but not implemented in the Universal SDK Demo app GUI).



In the screen shot above:

- The connected device is an Augusta card reader sending the **Get Extended Firmware Version** command, which is **78 31** for Augusta card readers communicating in HID mode (the same command is **52 31** for Augusta readers using USB-KB mode).
- The user has entered the command **52 31** in the text box (note: spaces don't matter).
- Above the box, the **Wrap as NGA** checkbox has been checked, which tells the USDK Demo app to create an NGA-format protocol wrapper around the command before sending it to Augusta.
- As a result, when the user executes **Send Data Command**, the actual byte string that gets sent over the wire is **020200783149a903**.

In this example, the response from Augusta is:

```
022200064944205445434820417567757374612055
53422d4849442056312e30332e3031352dd903.
```

The response contains the ASCII string " ID TECH Augusta USB-HID V1.03.015," wrapped in an NGA protocol wrapper.

9.6. Get Key Status

This command exists under the **Device** node of the commands tree for most ID TECH devices. Run it to see which cryptographic keys, if any, have already been injected into the device. Note that the presence of a key doesn't necessarily mean the device is in an encrypting mode unless it is an SRED device. Users may still have to turn on encryption with the **Set Encryption Control** command found under **Device.**

Note: After encryption is enabled, it cannot be disabled.

Note that there is no dedicated **Get Key Status** command for the Spectrum Pro. Instead, use the **Poll Card Reader** command (under **Device**), which returns six bytes of status information, with eight bit-flags per byte, including information not only on key status but PIN Pad connection status, tamper status, and more.

9.7. Get ICC Status (ATR)

Users can query a seated chip card for ATR information (Answer to Reset). The exact command may differ slightly from product to product. For example, if an Augusta unit is connected, use the **Power On ICC** command under **ICC** and check the **Results** panel. The USDK Demo app should display a message similar to:

```
ATR:494420544543482041756775737461205553422D4849442
056312E3031
ICC Powered On successfully
```

For a Spectrum Pro unit, use the **Get ICC Status** command under **ICC.** The USDK Demo app should display a message similar to:

```
ICC Powered On successfully
ICC Reader Status : [ICC Powered] [Card Seated] ICC
    Status : 21030000000a00000000
ATR : 3b6800000073c84013009000
```

9.8. Get Terminal Data and Set Terminal Data

This command returns a block of TLV data corresponding to terminal settings stored in the device. These settings, and their significance, are discussed in greater detail in the Which Terminal Settings Am I Allowed to Change? article on the ID TECH Knowledge Base. Consult that article to learn which TLVs are considered major and should not be changed versus which TLVs are considered minor and can be configured to tailor a device to meet specific needs.

Use the **Save Terminal Data** command to save TLVs (as a block) after making changes to various values. This is generally a one-time setup event that users do not need to repeat after a unit goes into production (although in practice, users may change values dynamically if needed).

Tech Support may ask users to provide the output of the **Get Terminal Data** command in certain situations. After running this command, the **Results** panel displays a complete listing of TLVs.

9.9. Save AID, List AID, and Load Default AID

The **List AID** command under the **EMV** node enumerates all Application Identifiers known to the card reader and displays them in the **Results** pane, regardless of which AIDs are present in a given chip card.

Conversely, the Universal SDK Demo app can load AIDs into a reader by means of the **Load Default AID** command. Using the **Load Default AID** command loads a hard-coded list of AIDs. This is a potentially time-consuming command; allow 10 seconds or more for it to complete.

To force a reader to load a specific AID and its associated TLV data, use the **Save AID** command and supply the necessary parameter data in the context-sensitive UI.

When attempting to perform EMV transactions in the Universal SDK Demo app, if transactions fail with a message of "Transaction Failed: No AID or No Application Data," run the **Load Default AID** command, then retry the transaction. The transaction should finish normally. If it doesn't, the card used for transactions may require a specific AID.

9.10. Save CAPK (and Related Commands)

The **EMV n**ode of the command tree contains several commands related to certification authority public keys (CAPKs), including **Save CAPK**, **Remove CAPK**, **Remove All CAPK**, **Load Default CAPK**, and **List CAPK**. Use these commands to load or query individual CAPKs as well as to list all CAPKs (using **List CAPK**) or remove all CAPKs. The Universal SDK (and Universal SDK Demo app) ships with over two dozen CAPKs, but make sure to check that none have expired (the 7th and 8th bytes of a CAPK give the expiration date as MM and YY).

Consult EMV Book 2 for detailed information about CAPKs.

9.11. Send APDU

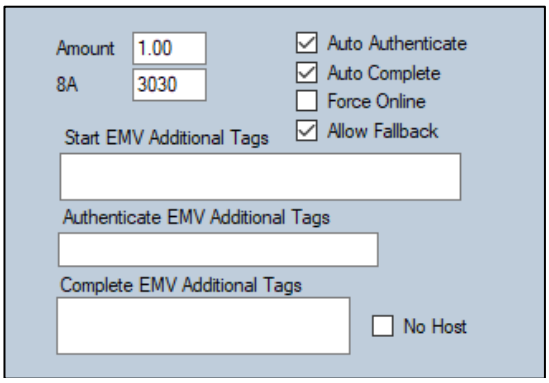
The Universal SDK provides convenience methods for exchanging APDUs with chip cards; this functionality is exposed for certain products (such as Augusta) in the Universal SDK Demo app via the **Send APDU** command under the **ICC** node of the command tree. Note that this low-level functionality is not currently exposed for every ID TECH product.

APDU-based communication with chip cards is an advanced subject. Ordinarily, users should not need this level of access to ICCs when working with the Universal SDK because the ID TECH EMV L2 kernel automatically handles application selection, data object list exchange, Gen AC requests, and other low-level card interactions without programmer intervention. Before attempting APDU-level I/O with a card, check to see if the desired functionality is already done via high-level-language routines in the regular Universal SDK API.

10. Executing EMV Transactions with Universal SDK Demo app

The Universal SDK Demo app exposes **Start EMV Transaction**, **Authenticate EMV Transaction**, and **Complete EMV Transaction** (as well as **Cancel EMV Transaction**) methods in the UI. Look under the **EMV** node of the command tree to find these commands and make sure to read about them in the SDK documentation.

Each of the commands listed in the above paragraph are context sensitive. Selecting any of them in the EMV list of commands changes the appearance context-sensitive area of the UI.



Optionally, enable **Auto-Authentication** and **Auto-Completion** by checking the appropriate checkboxes, or change the primary **Amount** of the transaction (the default is $1.00), and set the response code of Tag 8A (for example, test "unable to go online" by setting the value here to **5A 33**).

Individual text areas allow users to specify which additional TLV tags to receive after each transaction step (**Start**, **Authenticate**, or **Complete**). Note that any tags specified here override the normal set of tags the respective transaction steps return.

Also note that tags 5A (PAN) and 57 (Track 2) are usually provided by default in the data the **Start Transaction** command returns (though they may not be available in later steps). To see data for tags 5A and 57, leave **Auto-Authenticate** and **Auto-Complete** unchecked (note: these tags normally contain encrypted data).

10.1. Running a Contact-EMV Transaction

To run a contact-EMV transaction in the Universal SDK Demo app:

1. Make sure the terminal data configuration is set as desired (see Get Terminal Data and Set Terminal Data above).
2. Make sure to load any AIDs and CAPKs necessary for the transaction.
3. Insert a card in the reader.
4. Execute the **Start EMV Transaction** command.

IMPORTANT: When requesting **Additional Tags** using the text areas shown above, please note that users must retrieve this data manually with the **Retrieve Tags** command (under **EMV**), after the transaction finishes. Also note that transaction data, as a rule, is available for only 15 seconds after a transaction; for security reasons, such data is purged after that period of time. Therefore, execute **Retrieve Tags** promptly.

The Universal SDK Demo app can run all phases of an EMV transaction (and simulate an online authorization) while testing various CVMs, parameter options (for example: allow or don't allow fallback to MSR), and error conditions. In real-world applications custom program logic would need to pause at some point to go online to receive the actual authorization response from the acquirer or issuer (typically via calls to a web service). The Universal SDK Demo app can only simulate this portion of the transaction cycle.

The Universal SDK Demo app itself cannot actually "call out" to a web API; users must build this functionality into apps on their own.

For more information, see the ID TECH white paper, "EMV Transactions with the Universal SDK," available on the ID TECH Knowledge Base.

10.2. Start Transaction Commands

The following commands initiate transactions on devices for the listed platforms.

10.2.1. NEO 2: Contact Start Transaction (60-10)

The **60-10** command starts a new contact EMV L2 transaction (ICC + MSR) or an MSR-only transaction for NEO 2 devices.

10.2.2. NEO: Contact Start Transaction (60-10)

The **60-10** command starts a new contact EMV L2 transaction (ICC + MSR) or start MSR only transaction for NEO 2 devices.

10.2.3. AR: Activate Transaction (02-01)

The **02-01** command initiates a transaction with a supported card on AR devices.

10.2.4. AR: Secure Enhanced Activate Transaction (02-05)

The **02-05** command should be used when a transaction must be completed with any supported contactless EMV, contactless Magstripe card, contact EMV, or Magnetic Stripe card while “Secure

Mode" is in effect (nevertheless, this command also works in non-Secure Mode.)

10.2.5. AR: Enhanced Activate Transaction (02-20)

Use this command when you want greater control over transactions. The command performs the same function as the Activate Transaction command with options set using four bytes of transaction bit flags. Use the Enhanced Activate Transaction to:

- Specify which Interfaces to use when polling for a card
- Request a fallback transaction if an Activate Transaction or Enhanced Activate Transaction command returns a fallback status
- Control User Interface related features that are normally handled by the Reader. Select the desired method of communication: normal (blocking), or event-driven (non-blocking).

10.3. Setting Auto-Poll and Poll on Demand

The **Set Poll Mode (01-01)** command allows the terminal to set the reader's polling mode. The reader functions in one of two polling modes: Auto Poll or Poll on Demand. The value is saved in nonvolatile memory, so developers must only send this command to change between modes.

Note that the **Set Poll Mode** command has different parameters depending on the protocol each device uses (that is, NEO, NEO 2, and AR devices require passing different parameters to set the desired Poll Mode.

ViVOpay products operate in Poll on Demand Mode by default. Use the Poll on Demand Mode when to have the reader to poll for cards only when the terminal makes requests. In this mode the reader remains in the idle state with the RF field off until it receives an Activate Transaction (02-01 and 02-40) command. After the transaction is completed (or the reader times out while polling) the reader returns to the idle state. This mode allows the terminal to send data to the reader before the card data is read, as required for EMV transactions.

In Auto Poll Mode, the RF field is always active and the reader continuously polls for the presence of a contactless transaction. There is no requirement for the terminal to initiate a transaction. When a supported contactless MAGSTRIPE card is detected, the track data can be sent out on the magstripe interface (if the ViVOpay unit supports it) or retrieved using the Get Transaction Result (03-00 and 03-40) command. The Auto Poll Mode is required in environments where the reader is connected to a POS terminal via the terminal's magstripe interface.

10.4. Burst Mode

In Burst Mode (which requires the reader to be in Auto Poll Mode and is useful only for MSR data; see note below), the reader sends a data frame to the terminal each time it successfully reads a card. The reader continues polling for supported RF cards. Whenever the reader detects a card in the RF field, it tries to read the card data.

If the read operation is successful, the reader sends a "card payload" frame that contains the status, application type, card data, and CRC to the terminal through its serial port. Detailed information on the frame format is given in the sections ahead. The terminal does not have to send any command or data to the reader.

Note: The reader must be in auto poll mode for to use Burst Mode. Setting Burst Mode on for other configurations can lead to unexpected results.
Burst Mode is intended to be used with magnetic stripe card data only.

Enable Burst Mode with the **Set Configuration** command and the FFF7 tag. There are two options for Burst Mode: **Always On** (FFF7 = 01) and **Auto Exit** (FFF7 = 02). When the reader is in **Burst Mode Always On**, it ignores **Activate Transaction** and **Get Full Track Data** commands and remains in Burst Mode. When **Burst Mode Auto Exit** is enabled, the reader ends Burst Mode (FFF7 = 00) and processes these commands. Burst Mode then remains off until it is reactivated with a new **Set Configuration** command with tag FFF7 set to 01 or 02.

See the NEO or NEO 2 Interface Developer's Guide for more details about Burst Mode.

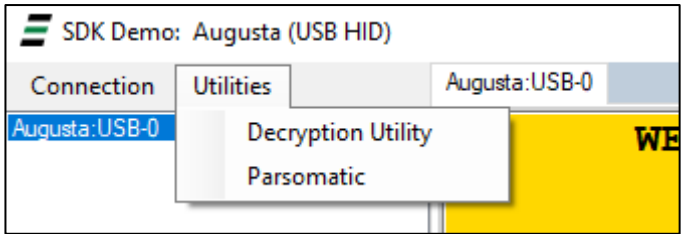
10.5. Tips

Users may find the following tips helpful when using the USDK Demo app to run transactions.

- Before running **Start Transaction**, make sure to load default AIDs (or, at a minimum, load the specific AID the card needs).
- Check to make sure the terminal Configuration is set properly for the anticipated environment (for example, chip-and-PIN environments may require the use of a different configuration than chip-and-signature). Check this area when experiencing CVM-related issues.
- Check to make sure any needed CAPKs are loaded; if transactions fail at the level of SDA, DDA, or CDA—which is to say the offline data authentication phase—this is something to check.
- Be aware that some TLVs will contain encrypted data, but only if a reader has been injected with the necessary keys, and only if encryption has been turned on. Consult "ID TECH Encrypted Data Output Formats" for information on which tags contain encrypted data and which do not. Obtain the latest version of this document from the Knowledge Base.
  NOTE: If a gateway needs encrypted Track 2 data conforming to particular specifications, investigate the use of ID TECH tags DFEF4B, DFEF4C, and DFEF4D. These are described in Tech Note 011: "Tags for Obtaining Encrypted Track Data," available for download at the Knowledge Base.
- To control the behavior of the kernel with respect to authorization response codes, investigate the use of ID TECH proprietary tag DFEE1B. Find information about this tag, and other proprietary ID TECH tags in the "ID TECH TLV Tag Reference Guide," available for download at the Knowledge Base. Also see the Knowledge Base article Why is Tag 8A Giving a Z3 Response Code?
- Remember that tags 5A and 57 are available after **Start Transaction** but may not be available after **Authenticate Transaction** or **Complete Transaction**. These tags contain sensitive data that will be encrypted.
- When requesting additional tags in any transaction step, remember to call **Retrieve Tags** to obtain them within 15 seconds after the transaction phase in question.
- When requesting additional tags in any transaction step, those tags (and only those tags) will be returned for that step.

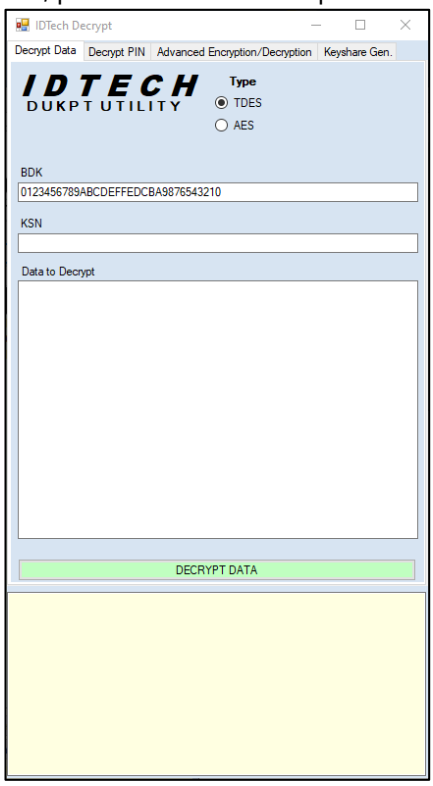
11. Demo App Tools

The USDK Demo app offers (or links to) two useful ID TECH tools: the ID TECH DUKPT utility, and Parsomatic.



11.1. Decryption

Clicking the Decryption button in the USDK Demo app launches the ID TECH Decryption tool, allowing users to decrypt data, PIN, plus other available options.



11.2. Parsomatic

Launch the ID TECH Parsomatic page by clicking the **Parsomatic** button in the USDK Demo app (or by going straight to the Parsomatic page in a browser). The Parsomatic tool parses ID TECH card-reader data streams from AR, GR, NGA, NEO I and NEO II devices. You can also launch Parsomatic directly in your web browser.

To use Parsomatic, paste the desired data into the text area and press **Enter**. If the data consists entirely of TLVs, check the **TLVs only** checkbox.

Parsed data appears underneath the text area. Hover the mouse cursor over color-coded areas to see an explanation of the hex values. Optionally, use the tag lookup feature (immediately below) to look up tags at any time.

12. For More Information

For more information on the Universal SDK, first obtain the specific SDK build for a particular device and operating system from the appropriate product page on the ID TECH Knowledge Base, then consult the **docs** folder of the installed SDK.

EMV Transactions with the Universal SDK, available on the Downloads page at the ID TECH Knowledge Base, is a good starting point for learning about EMV in general as well as the steps involved in carrying out EMV transactions with the Universal SDK.

To open a support ticket, use the Customer Support Portal or contact Tech Support via email at support@idtechproducts.com (emailing this address automatically opens a support ticket).