TECHNOLOGY SOLUTIONS UK LTD
part of HID

ASCII PROTOCOL LICENCE KEY SAMPLE APPLICATION (.NET DESKTOP)

www.tsl.com
Design • Development • Manufacture

CONTENT

Overview

This document describes the ASCII Licence Key Sample application provided as part of the ASCII 2 Windows Desktop SDK

History

Version	Date	Modifications
1.0	20/11/2014	Document creation

INTRODUCTION

The ASCII Protocol Licence Key Sample application was developed to provide developers with examples of using the .Net API to command devices that support the Technology Solutions ASCII 2 protocol. This sample demonstrates using the Licence Key command to enforce that an application can only be used with licenced readers. The Licence Key command was added in ASCII Protocol v2.2.

OVERVIEW

The licence key command is used to store or read a text value of up to 255 characters to or from a connected reader. The licence key command can simply be used to store a value in the reader (e.g. 'password') that the application can subsequently check. A more secure method is two combine a secret with a unique value from the reader into a one way function like a hash function. This makes the licence key specific to each reader and computationally difficult to determine a licence key for the same application for a different reader.

The version information command returns various properties about the connected reader. Included in these properties are the serial number and Bluetooth address, these are both read only and unique to each reader. The properties also include the supported ASCII Protocol version which can be used to determine if the licence key command is supported. Alternatively the reader will simply return an error response with command not supported.

This sample combines the serial number of the connected reader with a secret value and the company name and calculates an MD5 hash. The output of the hash is written to the reader using the Licence Key command as a base 64 value. A *LicenceKey* class is provided to compute and verify the hash value. By using a hashing function any variation in the input value (either reader serial number or other inputs) causes a vastly different hash output. The inputs to the hash are also computationally difficult to determine given the output (a desired property of a good one way function like a cryptographic hash). This should prevent a casual operator from being able to fake licence key values into other readers, with different serial numbers and Bluetooth addresses, even knowing an existing licence key value for a particular reader.

USING THE APPLICATION

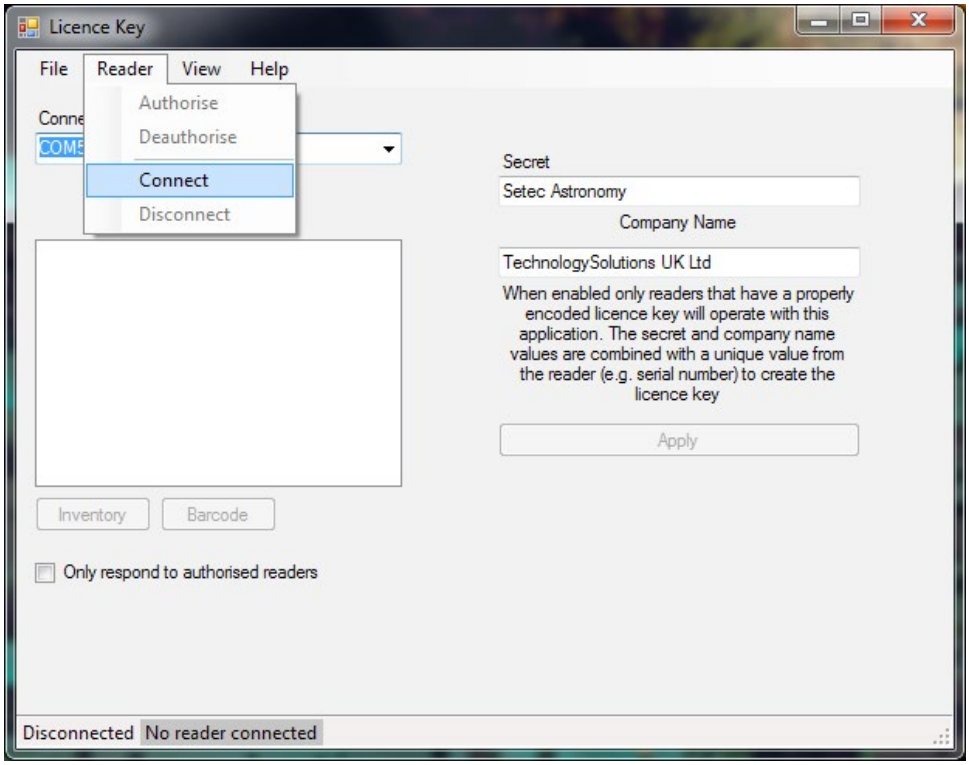

FIGURE 1: The main screen

The main screen shows the status of the connected reader and has an output window for messages.

After selecting the connection name using the combo box The Reader>Connect and Reader>Disconnect menu items control the connection to the reader. The authorisation and connection status are displayed in the status bar.

The 'Only respond to authorised readers' checkbox changes the mode of the application. When unchecked the application will operate with the reader as normal. Trigger barcode scans and inventories are displayed in the message window. Using the 'Inventory' or 'Barcode' menu items will command a similar action in the normal way. When checked the application will only operate with authorised readers. Trigger barcode scans and inventories still perform but the response is ignored by the application. The 'Barcode' menu item is disabled and the "Inventory' menu item returns an appropriate error.

The View>Clear Messages menu item clears the messages in the messages window.

The Reader>Authorise menu item will authorise the current reader for use with the current reader by programming an appropriate licence key into the reader.

The Reader>De-authorise menu item will delete any licence key from the connected reader so that it is no longer authorised for use with the application.

SETTINGS

The settings user control on the right of the page simply shows the two strings that are used as the values unique to the application for calculating the hash. By changing these values and clicking apply after authorising a reader you can see that the hash value based on these strings changes for the same reader and the reader is no longer authorised.

AUTHORISATION

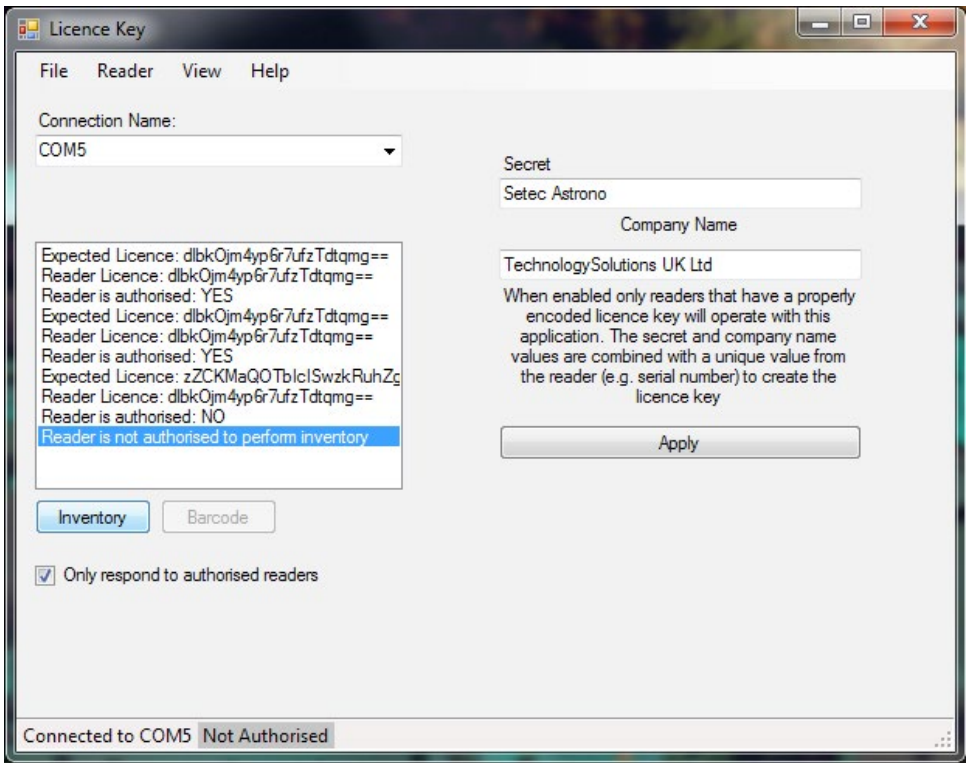

FIGURE 2: States of authorisation

As shown above when the hash value calculated from the reader's serial number and secret matches the value stored for the licence key the reader is authorised. When the secret is changed or the reader is changed the hash value changes and the message window reports the reader is not authorised to perform inventories when the 'Inventory' menu item is attempted.

CODE DESCRIPTION

The application is based around the *MainForm* class which is created in the usual manner from a *Program* start-up class. The logic for the *MainForm* is delegated to the *MainViewModel* class. The *MainForm* also uses the *ConnectViewModel* to manage the reader connection. The *SettingsUserControl* uses a *SettingsViewModel*.

The *MainViewModel* communicates with the *ReaderService* via the *IReaderCommand* interface. This interface provides events when a transponder or barcode is received, provides an indicator whether a reader is connected and provides a method to execute ASCII commands.

The *ConnectViewModel* communicates with the *ReaderService* via the *IReaderConnect* interface. This interface provides methods to connect and disconnect from a reader.

The *SettingsViewModel* only uses a reference to the Settings class to edit the values used in the *LicenceKey* class to calculate the hash.

Within the *MainViewModel*: *AuthoriseReader* first performs a *VersionInformationCommand* and checks that the ASCII Protocol version is greater than or equal to version 2.2 (the minimum ASCII Protocol that supports the *LicenceKeyCommand*). If supported the reader *SerialNumber* is passed to the *LicenceKey.Compute* function along with the *Secret* and *CompanyName* properties to create the licence key value. The computed licence key is then set as a parameter of the *LicenceKeyCommand* along with the *Deletion.Yes* parameter. The *Deletion.Yes* parameter is specified as the existing key if any must be deleted to program a new licence key. The *LicenceKeyCommand* is then executed to program the licence key to the reader. If the command is successful the user interface is updated to show the reader is now authorised.

DeauthoriseReader simply executes a *LicenceKeyCommand* with the *Deletion.Yes* parameter set to delete any existing licence key. The user interface is updated to show the reader is no longer authorised.

The *VerifyAuthorisation* function is called whenever the *MainForm* Activated event fires. This verifies the authorisation status when returning from the *ConnectForm* or the *SettingsForm*. If not connected the function just sets *IsConnected* and *IsAuthorised* to false. If connected the function first performs a *VersionInformationCommand* to ensure the *LicenceKeyCommand* is supported and also obtain the serial number of the reader. The *LicenceKeyCommand* is then performed with no parameters to read the current licence key from the reader. The *LicenceKey* class is then used to compare the reader licence key to the computed value to determine if the reader is licenced.

FURTHER INFORMATION

More information can be found on the Technology Solutions website. The product downloads section of each product requires a free, one time, registration. See “Product Downloads” of the following products to download the document describing the ASCII Protocol and also user manuals for the products.

http://www.tsl.uk.com/products/1119-uhf-rfid-reader-for-the-motorola-mc55-65/
http://www.tsl.uk.com/products/1126-desktop-uhf-rfid-reader-with-usb/
http://www.tsl.uk.com/products/1128-bluetooth-handheld-uhf-rfid-reader/
http://www.tsl.uk.com/products/1153-bluetooth-mini-uhf-rfid-reader/

If you have any questions please contact support@tsl.uk.com

ABOUT

ABOUT TSL®

Technology Solutions UK Ltd (TSL®), part of HID Global, is a leading manufacturer of high performance mobile RFID readers used to identify and track products, assets, data or personnel.
For over two decades, TSL® has delivered innovative data capture solutions to Fortune 500 companies around the world using a global network of distributors and system integrators. Specialist in-house teams design all aspects of the finished products and software ecosystems, including electronics, firmware, application development tools, RF design and injection mould tooling.
TSL® is an ISO 9001:2015 certified company.

ISO 9001: 2015

CONTACT

Address:	Technology Solutions (UK) Ltd, Suite A, Loughborough Technology Centre, Epinal Way, Loughborough, Leicestershire, LE11 3GE, United Kingdom.
Telephone:	+44 1509 238248
Fax:	+44 1509 214144
Email:	enquiries@tsl.com
Website:	www.tsl.com

ABOUT HID GLOBAL

HID Global powers the trusted identities of the world's people, places and things. We make it possible for people to transact safely, work productively and travel freely. Our trusted identity solutions give **people** convenient access to physical and digital **places** and connect **things** that can be identified, verified and tracked digitally. Millions of people around the world use HID products and services to navigate their everyday lives, and billions of things are connected through HID technology. We work with governments, educational institutions, hospitals, financial institutions, industrial businesses and some of the most innovative companies on the planet. Headquartered in Austin, Texas, HID Global has over 4,000 employees worldwide and operates international offices that support more than 100 countries. HID Global is an ASSA ABLOY Group brand.
For more information, visit www.hidglobal.com.