

# Product Integration Topic - Augusta QuickChip Keyboard fallback behavior

## Premise:

This article focuses on understanding what needs to be addressed to ensure that these brand test cases can be passed properly, and educate you on what tags need to be configured for your production use cases to properly address fallback/

**Resources:** (Have these prepared beforehand)

IDTECH Parsomatic Tool: <https://www.idtechproducts.com/hosted-files/tools/parsomatic.html>

IDTECH Knowledge Base to Universal Demo: [Universal SDK - downloads](#)

## Scope:

This article covers the functionality of the Augusta QC KB in regards to fallback behavior.

Behavior such as no matching application, bad chip inserts, and swiping a chip card should be considered in these use cases.

For output tags and other related items, you can consult one of the handy guides here:

[-FAQ: Augusta Quick Chip Keyboard \(QCKB\)](#)

## What you need to know (at a high level):

- IDTECH has defined several proprietary tags that control the device's fallback behavior and output format.
- This article is mainly applicable to keyboard emulation - the Augusta QC KB does NOT communicate with the host device
- The Augusta is functioning as a keyboard and cannot have any dialogue with the host, the written application must be 'smart'
- The integrating party must be able to receive the input tags / error codes and interpret them in the application, and handle them as appropriate depending on test case (ie: show 'insert card', show 'no matching AID' and enable MSR)
- Any test cases that require Cardholder Selection can be waived. The Augusta in terminal configuration 5C does NOT do cardholder selection.
- Your terminal configurations should be versioned and shared with all parties involved in the certification, to ensure all members are on the same page.
- All options for reporting can be enabled or disabled depending on what is needed.

## Implementation:

There are several TLVs you'll need to set in the terminal configurations to address and handle fallback.

Tag	Length	Definition	Description / Usage
DFED0A	1	Outputs Fallback Reason	0: Switch is off, does not output fallback reason. 1: Switch is on, does output fallback reason
DFF65	1	Controls Error Reporting	0: Switch is off, does not output errors. 1: Switch is on, does output errors
DFF62	1	Controls whether MSR card with a chip on it can be swiped	0: Allows swipe from chip card. 1: Disallows swipe from chip card.
DFF7D	1	Controls the number of times the reader will allow another insert after a bad insert	3 times is default.
DFF7E	6	Controls the status messages returned by the device while the device is in fallback.	<b>Leave as: DF EF 7E 06 50 01 50 05 50 36, do not change once added.</b>

## Detailed Explanation of Tags:

DFED0A - Quiet Mode for fallback reason error reporting in return swipe







```

80 // Card Encode Type (1 byte, 2 characters)
1F // Track Status (1 byte, 2 characters)
38 // T1 Length -> 56 (8*7) (1 byte, 2 characters)
28 // T2 Length -> 40 (8*5) (1 byte, 2 characters)
00 // T3 Length (1 byte, 2 characters)

// Track Data (ASCII) (60 characters)

A19B%601197*****00
05^DEBIT/IMAGE 01^23
12*****?

B32D71FE69D36E4F // Encrypted T1 (56 bytes, 112 characters)
A06DE869CFC5BD1A
4E9AB526D9A931A5
4CC514318912924A
1AEC0F29EEBB67AE
D3A3C1D90E9603CE
37494DC86749EC27

48024441E96881A1 // Encrypted T2 (40 bytes, 80 characters)
9035A7A6C76BD037
A8030F17606FB691
4401C5C3975D0A65
9617C98A708BF712

00000000000000000000 // Hash T1 (20 bytes, 40 characters)
00000000000000000000

00000000000000000000 // Hash T2 (20 bytes, 40 characters)
00000000000000000000

36323554373030313534 // Device Serial Number (10 bytes, 20 characters)

62994900000000000060 // KSN (10 bytes, 20 characters)

076D // LRC / Checksum (2 bytes, 4 characters)

03 // ETX (1 byte, 2 characters)

```

-----

For the MSR portion only, following our enhanced format, we get:

$2+2+2+2+2+60+112+80+40+40+20+20 = 382$

Going back to length what we were looking at for this TLV, we have **395**.

$2 + 4 + 382 + 4 + 2 = 394$  total characters

The missing character comes from the carriage return character (in the fallback, it's between T1/T2)

**Adding on the carriage return, we have 395.**

For the non-encrypted case, I imagine that is where the missing character can be found as well (most likely a carriage return or some other character that is counted)

9F390180

DFEE23 25 (37)

;5413330089020029=2512201062980790? (36 characters + some possible carriage return would get you 37)

