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 and Reference - 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
DFEF65	1	Controls Error Reporting	0: Switch is off, does not output errors. 1: Switch is on, does output errors
DFEF62	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.
DFEF7D	1	Controls the number of times the reader will allow another insert after a bad insert	3 times is default.
DFEF7E	6	Controls the status messages returned by the device while the device is in fallback.	**Leave as: DF EF 7E 06 50 01 50 05 50 36, do not change once added.**

Detailed Explanation of Tags:

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

Tag DFED0A controls the return of a "DFEE25" return status code in a fallback swipe. When the tag value is 1, the return swipe will always contain the type of fallback that was done. This is typically needed for an implementation that needs to identify the type of fallback from just the swipe data alone. You can see the types of DFEE25 below.

To control it, you can add **DFED0A** to your terminal configs. Set to 00 for no reporting, 01 for error reporting. This value by default is 0.

- **DFED0A 01 01 will allow reporting fallback reason**
- **DFED0A 01 00 will disallowreporting fallback reason**

DFEF62 - Not allowing MSR swipe with chip on card:

Tag **DFEF62** controls whether an MSR card with a chip on it can be swiped. If this value is set to 1, the Augusta will not output any data when a chip card is swiped. This value by default is 0.

- **DFEF62 01 01 will disallow MSR swipe on ICC, only allowing swipe in fallback situations.**
- **DFEF62 01 00 will allow MSR swipe on ICC regardless of situation.**

DFEF65 - Quiet Mode for error reporting:

To enable it, you can add DFEF650100 to your terminal configs. Tag DFEF65 controls the output of a "DFEF61" error code when the device has a chip card inserted. When the tag value is 1, the device will output DFEF61 each time the card is inserted badly. You can see the types of DFEF61 below.

- **DFEF65 01 01 will allow reporting**
- **DFEF65 01 00 will disallow reporting**

DFEF7D - **Number of Fallback times**

Tag **DFEF7D** controls the number of times DFEF61 will be thrown (default is 3). This controls the number of times the Augusta can ask for a chip insert before terminating the transaction.One of the AMEX test cases requires 3 inserts before fallback, and IDTECH has confirmed with FirstData / Vantiv that IDTECH's method listed here will be valid. The Augusta will enter fallback mode each time DFEF61 is thrown.

- **DFEF7D 01 03 will allow the device to take 3 bad chip inserts before terminating the transaction session**
- **DFEF65 01 01 will allow this behavior once**

DFEF7E – Status codes that will trigger fallback

Tag DFEF7E is a configuration tag (for contact EMV terminal settings) that determines which "status codes" (ICC response codes) will trigger fallback. For example, an empty candidate list will cause the return of code 0x5005 and the transaction will terminate. If you put 5005 in DFEF7E, the reader will carry out MSR fallback behavior instead of just terminating. You would specify DFEF7E025005 in the contact terminal settings.

NOTE: You can specify up to 32 ICC response codes (64 bytes of data) into DFEF7E.

Here is an example of the below list of additional TLVs that you will need to add to your existing terminal configuration:

> *// Output Fallback reason, show error reports, allow swipe from chip cards, reader will fall back 3 times*
>
> *DF ED 0A 01 01 DF EF 65 01 01 DF EF 62 01 01 DF EF 7D 01 03 DF EF 7E 06 50 01 50 05 50 36*

A complete terminal configuration (with the relevant TLV blob) is shown below as an example:

> *9F 02 06 00 00 00 00 01 00 5F 36 01 02 9F 1A 02 08 40 9F 35 01 21 9F 33 03 60 28 C8 9F 40 05 F0 00 F0 A0 01 9F 1E 08 54 65 72 6D 69 6E 61 6C 9F 15 02 12 34 9F 16 0F 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 9F 1C 08 38 37 36 35 34 33 32 31 9F 4E 22 31 30 37 32 31 20 57 61 6C 6B 65 72 20 53 74 2E 20 43 79 70 72 65 73 73 2C 20 43 41 20 2C 55 53 41 2E DF 26 01 01 DF 10 08 65 6E 66 72 65 73 7A 68 DF 11 01 00 DF 27 01 00 DF EE 15 01 01 DF EE 16 01 00 DF EE 17 01 05 DF EE 18 01 80 DF EE 1E 08 D0 9C 20 D0 C4 1E 16 00 DF EE 1F 01 80 DF EE 1B 08 30 30 30 31 35 31 30 30 DF EE 20 01 3C DF EE 21 01 0A DF EE 22 03 32 3C 3C DF EF 4B 03 7F 00 00* ***DF ED 0A 01 01 DF EF 65 01 01 DF EF 62 01 01 DF EF 7D 01 03 DF EF 7E 06 50 01 50 05 50 36***

Augusta QC KB Response Codes / Returns:

DFEF61: Status of Insert

Value	Definition
F2 20	Insert ICC again / Swipe (used for all other fallback)
F2 21	Prompt Fallback (used for case of no matching AID)
F2 22	Counter reached/almost exhausted

DFEE25: Status of Swipe

Value	Definition
02 03	(Good ICC card) Transaction has reversal; declined.
30 12	ICC error, transaction exit

Common Scenarios: (Work In Progress)

Handling 'No Matching AID' scenario:

If the above configurations are set - on the case of no matching AID, the device will immediately enter fallback mode.

Here's a sample flow:

On insert:

> *DFEF6102F221*

The device will begin to flash blue and beep rapidly. Upon card swipe:

> ***9F39*** *01 80* ***DFEE23*** *82 01B702AA01801F442800A19B%*651000******0158^CARD/IMAGE 11 ^1712**********?*
> *2411965A59042C3B8FA8BECF3D2C5EBFDFB621433B19DBC30CC8A8813237BA23D3865AB53D67FC5DFB7CD3A23DD2AF8*
> *593FE0D09993D39DA22FF8756BEBE55E2256EDBA3275A5C63DC41031DE127153516A37034811078F57C93EB6B891D8E8A3*
> *B958480CE9D6EFEFCA9D4615FF1C62D0000000000000000000000000000000000000000000000000000000000000000*
> *0000000036323554373030313534629949000000000000356EBA03*

The fallback MSR data will be shown in **DFEE23**, and will follow the IDTECH Enhanced Encrypted format, and can be parsed out for the PAN, expiration, etc.

You can see that in the above scenario, **DFEF61** is the only TLV returned by the Augusta. As there was no issue with the chip, IDTECH specified a different response code to allow you to address this particular case.

Handling the 'Bad insert' scenario (review 'Fallback Scenario' first):

If the above configurations are set - on the case of a 'bad insert', you'll receive the response code: '**DFEF6102F220',** and the device will enter fallback mode.

On the final (failed) insert, the reader will respond with **DFEE25023012** and terminate the transaction.

Here's a sample flow if **DFEF7D** was set to 03 (so **DFEF7D**0103) - default will be 3

- 1st insert bad ICC

DFEF6102F220 (prompt to INSERT ICC OR SWIPE AGAIN)

- 2nd insert bad ICC

DFEF6102F220 (prompt to INSERT ICC OR SWIPE AGAIN)

- 3rd insert bad ICC

DFEF6102F222 (prompt to INSERT ICC OR SWIPE AGAIN, counter almost exhausted) 05 Jul 2017 (per v18 update)

- 4th insert bad ICC

DFEE25023012 (ICC error, transaction exit)

You can configure **DFEF7D** to adjust how many times the device will allow swipe before termination.

Note how there is a difference in the status codes before the transaction edits (response code is F222 instead of F220)

This would let you (on your end) accept F2 22 and then return "Please Swipe; bad card" or something to that effect on your implementation. This is supported in firmware version v1.00.018.

The output from the device after a successful insert will be TLV data and can be treated as a normal chip insert.

The output from the device after a swipe may look like this:

> ***9F39*** *01 80* ***DFEE23*** *82 01B702AA01801F442800A19B%*651000******0158^CARD/IMAGE 11 ^1712***********?*
> *B58E0EFC80C71F8D8A4234765C2F3F5F90FBA991CDDB13CE8A6BBBF63BE9CC6CE165BB726910CA7C8CC1CA8D0C18179*
> *C87F0BA5EDF511496494A86ED369D9D487020C4D97A021C40E800DFFAAFEE3DC07F5E031798C542386983CB1460A51B187*
> *3AE5D27F93D7AD0884ACA2B7EA82DAF0000000000000000000000000000000000000000000000000000000000000000*
> *0000000036323554373030313534629949000000000034140C03*

The fallback MSR data will be shown in **DFEE23**, and will follow the IDTECH Enhanced Encrypted format, and can be parsed out for the PAN, expiration, etc.

The value of terminal entry mode **9F39** should be captured and presented to your processor.

Length of tag DFEE23:

Due to the Augusta returning ASCII embedded in the return MSR data (as how our other readers function), we had to figure out the best way to move forwards.

Long story short - the length you see is a character count rather than the actual length, including a carriage return. See proof below on this.

Original Swipe (MSR Fallback) // needs to be replaced 10 Aug 2017

DFEF6102F2209F390180DFEE2382018B027E01801F382800A19B%601197****0005^DEBIT/IMAGE 01^2312**********?
*B32D71FE69D36E4FA06DE869CFC5BD1A4E9AB526D9A931A54CC514318912924A1AEC0F29EEBB67AED3A3C1D90E9603CE37494DC86749EC27
48024441E96881A19035A7A6C76BD037A8030F17606FB6914401C5C3975D0A659617C98A708BF712000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000363235543730303135346299490000000000060076D03

Parsed Out:

DFEF61 02 F220
9F39 01 80

DFEE23 82 // Following BER TLV Structure, we can't have tags greater than 80. **82 indicates we will use next 2 bytes to denote the length.**

01 8B **//** Evaluates to 395, which is a **character** count of the below. **This is where we break TLV structure since we have to deal with ASCII.**

------ ENHANCED ENCRYPTED MSR OUTPUT -----

02 (1 byte)

7E 01 (382) (2 bytes) **// Length of actual MSR Payload, which is everything below until LRC/Sum/ETX,**

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

0000000000000000000 // Hash T1 (20 bytes, 40 characters)
0000000000000000000

0000000000000000000 // Hash T2 (20 bytes, 40 characters)
0000000000000000000

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)

Update as of 20 Sep 2017 with FW version Augusta_V1.02.006.fm:

Summary: Customers request that the type of the fallback be reflected in the return payload.

BAD INSERT / NO POWER UP
DFEF6102F220 (on bad insert)

// Swipe data

DFEE25023002
9F390180DFEE238201DC02D001801F422800A39B%*3607******1426^CARD/IMAGE 23^**********************************?*;
3607******1426=**********************?
*DA781354F3BEBD9C6CF832257D07E1B7CAA49AC95444B80CFF352E31613DD5FE2D7DD1F81FBC40C94D41ADB4540273BE4141D1B9660B116E
698B15B2CB6F1887AC4E72BF55A74247E2F2E47962BC9157ABD2D2A571247BFA5836FC3E603AF095F6260392B62BC9EABC4AD5F0CCA3857200
000000000000000000000000000000000000000000000000000000000000000000000000003632355437303031303 5FF7299490000038000280ADA
03

NO MATCHING APP
DFEF6102F221 (on no matching aid insert)

// Swipe data

DFEE25025005
9F390180DFEE238201DC02D001801F422800A39B%*3607******1335^CARD/IMAGE 19^**********************************?*;
3607******1335=**********************?
*033E7ABA64463E58AC94EEA8B1D813A269CF1829BE83010681A49F24F51AD081FF56CE00130B3808CB9AE14B7F986C6B673689EB17E1DF8E4F
1E90A15805E8ED8902DDB9E3F510DC9F25BEC54BC176A303049243DBE587EA28563F800F8BAA6382F16E624BD244927F4E619F59C39E8900000
0000000000000000000000000000000000000000000000000000000000000000000000000363235543730303130 35FF72994900000380002900 9C03