Reference number of document: **ITSO/COR 2.1.4-6**

Title: Corrigendum to Version 2.1.4 of the ITSO Specification

Specification Part(s) affected by this note: TS 1000-6 (Version 2.1.4) TS1000-9 (Version 2.1.4)

Source of document: **ITSO**
ITSO – Chief Advisor Technology

Change Control Details

Version	Date	Comment
1	21/11/2012	Initial Publication

Document type: ITSO Specification Corrigendum
Document subtype: COR
Document template: ITSO 2

ITSO

Corrigendum 6 to Version 2.1.4

This Corrigendum notice identifies the corrections to the Parts as noted below:

Corrected Versions of the Specification may be identified by the indicator COR 6 which can be found on the title page under the ISBN Number. The last modified date in the footer of the front page will be 18th October 2012 (2012-10-18) in Parts 6 and Part 9.

Reason for the Corrigendum:

A Technical Note (0416D1) had been drafted and approved by Technical Committee for incorporation into the next version of the Specification. The content of this Technical Note is designed to resolve some issues of ambiguity which have been causing problems in the implementation of, in particular, HOPS to HOPS message transfers. As a result the Technical Committee of ITSO at its October 2012 meeting recommended that the content of TN0416D1, with a few minor modifications for consistency with the existing V2.1.4, should be retrofitted into Version 2.1.4 of the ITSO Specification as a Corrigendum; the remainder of the changes being left for the next version in TN0416D2.

Note that it is not intended that any previously certified POSTs should need to have their methods immediately altered. (The text from TN0416D1 has been adjusted so that existing POST implementations should remain compliant without change, except for messages where Data# would be greater than 255.)

It is assumed that each POST / HOPS combination already has an agreed methodology for ensuring consistent communications, and as an interim measure these may continue to be used until POSTs have been updated to fully comply. Any new POST certification, and this includes any re-certification to update POST software, shall however be expected to follow the content of this Corrigendum, where Data# = either 0 or 1.

The implementation of this Corrigendum is however mandatory for all HOPS to HOPS communications and they should endeavour to be compliant at the latest by end February 2013. From a certification perspective; existing HOPS certification will be updated following receipt of documentation in writing from the HOPS Supplier's Primary Contact confirming compliance to this Corrigendum 6. It is therefore expected that all HOPS implementations will have been made compliant by 1st March 2013.

The following matrix has been included to aid the reader in understanding the implications:

Communication between	HOPS Communication From / To	Requirement	Implementation Required by:
HOPS to HOPS	HOPS	Full Compliance to this Corrigendum	1st March 2013
HOPS to POST and POST to HOPS	Compliant POST	Full Compliance to this Corrigendum	When POSTs become available with full compliance.
HOPS to POST and POST to HOPS	POST not yet updated	Continue to Use existing agreed methodology.	Already in use

It can be derived from the above that the implications are greater for HOPS than for POST devices.

Until all POSTs in a scheme have been updated, it will be necessary for a HOPS to be able to continue to use the current methodology [or methodologies if more than one] in place for transmitting and received long messages to and from POSTs and also to be able to transmit using the requirements of this corrigendum to any POST which has been updated to comply.

Part 6:

New or changed text is shown in blue. Text shown in italics is for location and instruction purposes only. Deletions where shown are in red with double strikethrough.

*Amend **Clause 2.1.1** as follows:*

2.1.1 Message Code Format

Message codes shall be a two byte value (x = any value). ~~The most significant byte shall have special significance as follows~~

Table 1 - Message Code Format

~~b 7-4~~	~~b 3~~	~~b 2~~	~~b 1~~	~~b 0~~	~~HEX 2~~ **Code (HEX)**	**Hashed Message ?**	**Type of message**	**Message Group**	**Reference to individual Message Definitions in this document**
~~0000~~	~~0~~	~~0~~	~~0~~	~~0~~	00xx	No	Transaction Record data	ITSO shell, IPE Administration, Card issuer messages	Table 7
~~0000~~	~~0~~	~~0~~	~~0~~	~~1~~	01xx	No	Transaction Record data	Stored Travel Rights, CTA	Table 7
~~0000~~	~~0~~	~~0~~	~~1~~	~~0~~	02xx	No	Transaction Record data	ITSO ID, loyalty, create or amend IPE, journey record, Actionlist acknowledge	Table 7
~~0000~~	~~0~~	~~0~~	~~1~~	~~1~~	03xx	No	Transaction Record data	Reversals and refunds, miscellaneous, list match event records	Table 7
~~0000~~	~~0~~	~~1~~	~~0~~	~~0~~	04xx	No	Other Message Data	Exceptions and card transaction error messages (POST to HOPS)	Table 7
~~0000~~	~~0~~	~~1~~	~~0~~	~~1~~	05xx	No	Other Message Data	POST to HOPS queries	Table 113
~~0000~~	~~0~~	~~1~~	~~1~~	~~0~~	06xx	Yes	Other Message Data	HOPS to HOPS and HOPS to POST messages	Table 78
~~0000~~	~~0~~	~~1~~	~~1~~	~~1~~	07xx	RFU	~~Other Message Data~~	~~AMS – ISMS messages, which are defined in ITSO TS 1000-8~~ RFU	~~----~~
~~0000~~	~~1~~	~~0~~	~~0~~	~~0~~	08xx	As Required	HOPS to POST, POST to HOPS and HOPS to HOPS messages	Miscellaneous messages~~, note that some message types may use a hash mac~~.	Table 146
~~0000~~	~~1~~	~~0~~	~~0~~	~~1~~	09 xx	No	HOPS to POST /HOPS messages	Message control	Table 2
~~0000~~	~~1~~	~~0~~	~~1~~	~~0~~	0Axx	No	HOPS to POST /HOPS messages	Parameter tables	Table 99

~~0000~~	~~1~~	~~0~~	~~1~~	~~1~~	0Bxx	Yes	HOPS to POST /HOPS messages ~~using a h Hash mac~~	Parameter tables	Table 99
~~0000~~	~~1~~	~~1~~	~~0~~	~~0~~	0Cxx	As Required	HOPS to POST /HOPS messages	Inter-operability list, Capability list, Hotlist, Actionlist, Data Correction record.	Table 78
~~0000~~	~~1~~	~~1~~	~~0~~	~~1~~	0Dxx	No	HOPS to POST /HOPS messages	HOPS Response to POST queries	Table 113
~~0000~~	~~1~~	~~1~~	~~1~~	~~0~~	0Exx	No	User defined system specific messages	Messages are user defined, and it is the responsibility of the sender to ensure that the addressee can interpret the message. Both data format and content are user defined, excepting that the messaging system shall assume that "the native format data is of another data type", as defined in ITSO 1000-9 annex A, clause A.3.3.[1]	----
~~0000~~	~~1~~	~~1~~	~~1~~	~~1~~	0Fxx	Yes	User defined system specific messages	Messages are user defined, and it is the responsibility of the sender to ensure that the addressee can interpret the message. Both data format and content are user defined. For the formatting and processing of long messages refer to ITSO TS 1000-9, excepting that the messaging system shall assume that "the native format data is of another data type", as defined in ITSO 1000-9 annex A, clause A.3.3.	----
~~Codes other than 0000~~	~~x~~	~~x~~	~~x~~	~~x~~	>1000	RFU	RFU	RFU	----

[1] These messages may be used to return user defined audit registers.

*Amend **Clause 5.2 Message Codes Table 78** heading as follows:*

Group	**Transaction Type**	**HEX CODE single record per Data Block**	**HEX CODE multiple records per Data Block (i.e. using a hash)**
	Rows omitted for clarity		

And add the following note beneath the table:

Where the size of an Actionlist item would exceed the maximum size permitted for a normal message using the 0C03 message type, for example when an IPE_Fulfilment_Action or a TYP24_Reservation_Group is sent, then the 0603 message type which uses a hash shall be employed.

Amend ***Clause 9.1 Message Codes 08xx, Table 146*** *as follows:*

Noting that the row pertaining to the 0801 message becomes shaded in this TN.

Table 146 - Miscellaneous Messages

Group	**Table Type**	**HEX CODE**	**Hashed Message?**
Miscellaneous Messages	Embodiment Parameter Request Message	0800	No
	Supplementary Data Message (Hash/Mac)	0801	Yes
	CM or Shell status advisory message	0802	No
	Physical ISAM Installation Notification	0803	No
	Customer Media Holder Details request	0804	Yes
	Customer Media Holder Details response	0805	Yes
	IPE Fulfilment Action notification	0806	Yes
	Additional Shell Data	0807	No
	Embodiment Parameter Request Message	0808	No
	RFU	0809 – 08FF	----

Note 1: Data in messages shown shaded in Table 146 are coded using ASN.1 notation.

Add the following sentence at end ***of Clause 6.9.5.1 ManifestHash*** *as follows:*

Note: The ManifestHash Data Element is purely related to the Manifest message and should not be confused with the HASH Data Element used when generating the Seal in a Data Frame Trailer of long messages.

Part 9

*Amend **Clause 4.4.14** sixth bullet point as follows:*

— Hash (for use with large ~~user defined~~ messages).

*Amend **Clause 4.4.16** as follows:*

~~This element has two options:~~

- ~~Message codes that use the standard sealing format, which is defined as message codes outside the ranges 0x0600 to 0x06FF, 0x0B00 to 0x0BFF, 0x0F00 to 0x0FFF and also excluding message code 0x0801.~~

- ~~Message codes that use the hash sealing format – which is defined as message codes in the ranges 0x0600 to 0x06FF, 0x0B00 to 0x0BFF, 0x0F00 to 0x0FFF and message code 0x0801.~~

For message codes that use the standard sealing format:

> The Data Block Length (DB Len) element is a single element and defines the total number of bytes in the Data Block, when said Data Block is encoded in its native format. This overall byte count shall include the DB Len element itself.

For message codes that use the hash sealing format (see TS1000-6 Message Code Format): *......Remaining text omitted for clarity.......*

*Amend **Clause 4.4.17** as follows:*

4.4.17 Number of Data Elements (Data#)

~~This element has two options:~~

- ~~Message codes that use the standard sealing format, which is defined as message codes outside the ranges 0x0600 to 0x06FF, 0x0B00 to 0x0BFF, 0x0F00 to 0x0FFF and also excluding message code 0x0801.~~
- ~~Message codes that use the hash sealing format – which is defined as message codes in the ranges 0x0600 to 0x06FF, 0x0B00 to 0x0BFF, 0x0F00 to 0x0FFF and message code 0x0801.~~

~~For message codes that use the standard sealing format, and optionally for message codes which use the hash sealing format:~~

> ~~The Number of Data Elements (Data#) element defines the number of data elements within the Data object.~~
>
> ~~When expressed in transmission format, a comma separator shall delimit each of said elements.~~

~~For message codes that use the hash sealing format~~

> ~~The Number of Data Elements (Data#) element may be set to a value of 1, indicating that the data object contains a file.~~

~~When Data# is set to 1 indicating that the data object contains a file, and when expressed in transmission format, comma separators shall not be used to delimit individual elements within the data object.~~

When creating any user defined Data object that:

a) Is NOT comma separated set Data# = 1
b) Has commas separating every element set Data# = 0

When creating any Data Object that contains ITSO Defined Data Elements (including elements that contain user defined values) use commas separating every element (as defined in clause 4.3) and set Data# = 0, or to a value between 2 and 255 (the value being the number of data elements in the data object).

Note: Previous versions of this specification allowed Data# values in the range 2 to 255 for comma separated ITSO Defined messages. For backwards compatibility, messages destined for POSTs may require Data# to be in the range 2 to 255. (For the avoidance of doubt, this includes POSTs certified to this version [2.1.4] of the Specification prior to December 2012.)

When receiving any Data object:

a) Data# = 1 indicates that the Data object within the Data Block is not comma separated
b) Data# = 0 indicates that the Data object within the Data Block is comma separated
c) Any other value of Data# indicates that the Data object is comma separated and its value shall be treated as though it was set to zero (0).

Note 1: The Hash Data Element is part of the Data object (within the Data Block) and therefore if comma separation is used shall be preceded by a comma

Note 2: Previous versions of this specification allowed Data# values in the range 1 to 255. Backwards compatibility with previous implementations is maintained by the recipient treating values of Data# in the range 2 to 255 as if they are zero (0).

*Amend **Clause 4.4.20** as follows:*

4.4.20 Data

The Data object is a composite object of variable size, depending on the type and content of the data to be transmitted. ~~The Data object consists of a variable quantity of elements, the number of which is specified by the Data# element (see clause 4.4.17)~~. The content of the Data object is defined according to the Message Code. Refer to ITSO TS 1000-6 for the detailed content and description of this data element.

*Amend **Clause 4.4.20(a)** as follows:*

4.4.20(a) Hash

This data element is only present in messages ~~having message codes in the ranges 0x0600 to 0x06FF, 0x0B00 to 0x0BFF, 0x0F00 to 0x0FFF and message code 0x0801~~ using the Hash (see TS1000-6 Message Code Format). ~~Use of this element~~ It allows messages containing Data Frames longer than 1024 bytes to be Sealed and verified using the ITSO SAM (see ITSO TS1000-8).......*Remaining text omitted for clarity.......*

*Amend **Clause 4.4.27** as follows (noting the deletion of the footnote number 6):*

4.4.27 User defined Data Frames

The ITSO messaging system defined above may be used to transfer ‘user defined’ data in Class 2 Application Messages. Special Message Codes~~[6]~~ (see TS1000-6 Message Code Format) are defined to signal such usage, and the required data shall be contained within the Data object of a standard Data Frame.

~~Where the ITSO messaging system is used to transfer ‘user defined’ Data Frames, the Data object of said frames shall comply with the following rules:~~

~~— For transmission, all data shall be in ASCII format.~~

~~— Commas shall only be used as element delimiters, and not for any other purpose.~~

— ~~The number of comma separated elements in the Data object shall be specified in the Data# field.~~

~ End of Corrigendum ~