ITSO

Issuing Authority:	Owner:	Project:
ITSO	Technology at ITSO	Technical
Document number:	**Part Number:**	**Sub-Part Number:**
ITSO TS 1000	3	
Issue number (stage):	**Month:**	**Year:**
2.1.5	March	2025
Title:		
ITSO TS 1000-3 *Interoperable public transport ticketing using contactless smart customer media – Part 3: Terminals*		
Replaces Documents:		
ITSO TS 1000-3 2010-02 issue number 2.1.4		

Revision history of current edition

Date	ITSO Ref.	Editor ID	Nature of Change to this Document (or Part)
Feb 2003	DCI 100 (create 2.1)	PQ	Document created
May 2003		PQ	Amended after editorial review
June 2003		JC	Re - fit to ITSO TC 1000 template, issued as CD
June 2003		SLB	Edit to conform to ITSO TS 1000 format, filed as 2nd CD
July 2003		JW	Major revisions following committee review
July 2003		SLB	Minor editorial adjustment. Issue as 2nd CD
Oct 2003		JW / SLB	Incorporation of comments. Incorporation of global changes. Issue as 3rd CD.
Nov 2003		SLB	Editorial changes only. Issue 1st consultation draft.
Jan 2004		JC	Implement DRC changes.
Feb 2004		CS	Check/consolidate DRC changes.
Feb 2004		SLB	Clean up and format as final draft.
Mar 2004		SLB	Implement final changes and prepare for issue.
Oct 2006		MPJE	Updated to include ISADs following approval by DfT
Jun 2007		MPJE	Updated to Version 2.1.2 – no changes to text
Feb 2008		PRJ/CJS	Updated to include ISADs following approval by DfT
Apr 2008		MPJE	Final editing prior to publication
Dec 2009		PRJ	Updated to include ISADs following approval by DfT
Feb 2010		MPJE	Final editing prior to publication
Mar 2015		MJC	Updated to incorporate Corrigendum 8 to Version 2.1.4
Apr 2015		MPJE	Updated to incorporate Corrigendum 9 to Version 2.1.4
May 2024		AM	Draft publication of Version 2.1.5
Mar 2025		AM	Updated to include ISADs following approval by DfT.
Mar 2025		AM	Final Editing prior to publication of Version 2.1.5

Document Reference: ITSO TS 1000-3

Date: 2025-03-31

Version: 2.1.5

Ownership: ITSO

Secretariat: Technology at ITSO

ITSO Technical Specification 1000-3 – Interoperable Public Transport Ticketing using contactless smart customer media – Part 3: Terminals

ISBN: 978-1-3999-8707-3

Foreword

This document is a Part of ITSO TS 1000, a Specification published and maintained by ITSO, a membership company limited by guarantee without shareholders. The membership of ITSO comprises transport organisations, equipment and system suppliers, local and national government. For the current list of members see the ITSO web site www.itso.org.uk

ITSO TS 1000 is the result of extensive consultation between transport providers, sponsors, system suppliers and manufacturers. The Department for Transport (DfT) has also contributed funding and expertise to the process.

Its purpose is to provide a platform and tool-box for the implementation of interoperable contactless customer media public transport ticketing and related services in the UK in a manner which offers end to end Loss Less data transmission and security. It has been kept as open as possible within the constraints of evolving national, European and International standards in order to maximise competition in the supply of systems and components to the commercial benefit of the industry as a whole. In general, it promotes open standards but it does not disallow proprietary solutions where they are offered on reasonable, non-discriminatory, terms and contribute towards the ultimate objective of Interoperability.

ITSO has been established to maintain the technical specification and Business Rules required to facilitate Interoperability. It also accredits participants and interoperable equipment. ITSO is a facilitator of Interoperability at the minimum level of involvement necessary. It will not involve itself in any commercial decisions or arrangements for particular ticketing schemes; neither will it set them up nor run them. It will however "register" them in order to provide the necessary Interoperability services (e.g. issue and control of unique scheme identifiers, certification and accreditation, security oversight).

Consequently, adoption of this Specification for particular ticket schemes will be a matter for the commercial judgement of the sponsors/participants, as will the detailed Business Rules and precise partnership arrangements.

Contents

1. Scope

ITSO TS 1000 defines the key technical items and interfaces that are required to deliver Interoperability. To this end, the end-to-end security system and ITSO Shell layout are defined in detail while other elements (e.g. terminals, 'back-office' databases) are described only in terms of their interfaces. The Business Rules that supplement the technical requirements are defined elsewhere.

1.1 Scope of Part 3

This Part of ITSO TS 1000 defines the requirements on Point Of Service Terminals (POSTs) in order that such terminals are able to support the Interoperable Smart Customer Media environment defined by ITSO. These POST requirements are grouped as follows:

• POST to Media interface	(external interface)	clause 2
• POST to HOPS interface	(external interface)	clause 3
• POST to ISAM interface	(internal interface)	clause 4
• Human interface		clause 5
• Functional requirements of the POST software		clause 6

Only requirements that are pertinent to Interoperable Smart Customer Media usage and interfacing to other parts of the ITSO Environment are defined herein. These requirements shall be applied as an Interoperability layer over the basic specification of a ticketing terminal. The overall specification of such a device is outside of the scope of this document.

For the avoidance of doubt, the fact that a POST may be accredited as ITSO compliant does not mean that it is fit for purpose in any area other than its support for Interoperable Smart Customer Media usage.

1.2 Context of a POST within the ITSO Environment

Within the ITSO Environment, a POST is defined as one class of instances of equipment that allows transactions to be carried out with data entities held on the Customer Media.

Within the ITSO Environment, the following actors are typically POST users:

• Product Retailers	(Sale and loading of ITSO Product Entity (IPE) instances into an ITSO Shell)
• Service Operators	(Validation and use of IPE instances)
• Customer	(In unattended transactions)
• ITSO Shell Retailers	(Sale and loading of an ITSO Shell on to the Media)

POSTs provide the access mechanism to the Media platform to support the ITSO aim of permitting the interoperable use of a variety of ticketing products, retailed and used on a variety of platforms, from different issuers, on behalf of multiple product owners.

The range of equipment covered by the term 'POST' is wide and diverse. However, all POSTs shall have the following key attributes:

• they shall be able to read and write data to contactless Media that comply with the ITSO Specification

• they shall periodically exchange data with a HOPS system

- they shall physically contain an ITSO Secure Application Module (ISAM), which is unique to the POST and is not shared by other POSTs

Equipment classes that either do not contain a dedicated ISAM (e.g. Media viewer) or use shared ISAM resources (e.g. ISAM on a remote server) are outside the scope of this document.

1.3 Co-existence

Ticketing using ITSO Customer Media is likely to co-exist with other ticketing mechanisms within a given system. Thus the requirements specified herein for POSTs will, in many cases, co-exist with other functions on the unit.

POSTs shall be designed such that the ITSO functionality does not interfere with, or be compromised by, non-ITSO functionality. In particular:

- the POST may process other Customer Media or applications in whatever manner required by suppliers Support for such Customer Media or applications shall not compromise the ITSO implementation

- the ITSO implementation shall not adversely affect the reliability of the POST. If other ticketing modes are possible without ITSO Customer Media, then these may operate if the ITSO sub-system is non-functional

1.4 Configurability / future enhancement

ITSO specifies the minimal level of POST configurability to allow interoperable operation based on this version of the Specification. As it is highly likely that additional Media platforms, products and messages will be developed and incorporated into future versions of the Specification, it is strongly recommended that POST designers use an architecture that allows for such future enhancement, preferably by means of configuration parameters.

This potential need to allow enhancement of the POST's capability in the future should also be noted by parties involved in system procurement.

2. POST to Media interface

2.1 General

Within the ITSO Environment, the primary purpose of a POST is to read and write to contactless Media. As such, all POSTs shall provide a contactless interface that complies with:

- [ISO 14443-2]   Type A and B
- [ISO 14443-3]   Type A and B
- [ISO 14443-4]   Type A and B

The POST shall also support the proprietary 'Mifare Classic' contactless interface.

The POST shall provide a single target area that allows Media to be easily presented by users and be read and written to in a reliable manner. The target area design shall be such that POSTs are able to process Media with form factors other than that defined by [ISO 14443-1].

ITSO does not require that POSTs be capable of operating with multiple Media in the reader field. If two or more devices, with which a POST is capable of conducting a transaction, are placed in the reader field at the same time, then the POST shall instigate an interaction with the customer before conducting a transaction.

2.2 Platform support

All POSTs shall recognise the entire set of platforms defined in ITSO TS 1000-10. POSTs shall also implement and use the appropriate Anti-tear mechanisms as defined in ITSO TS 1000-10 for each platform type.

It is strongly recommended that POSTs use a software architecture that allows further platforms to be added by means of configuration parameters.

Where the POST supports non-ITSO platforms, then the inclusion of this capability shall not compromise the terminal's ability to process ITSO defined Media platforms.

A POST shall not require any manual input to select Media type, but shall do this automatically by issuing the required ISO/IEC 14443-3:2001 'polling' commands in a manner that allows detection of all defined Media platforms.

A POST shall poll for all defined platforms in an un-biased manner, with each platform type having the same likelihood of detection and selection within a given time period.

2.3 Application Family Identifier

ISO/IEC 14443-3:2001 provides for support of an Application Family Identifier (AFI) pre-selection mechanism.

ITSO does not mandate the use of AFI coding. POSTs shall not assume that Media use AFI coding, and shall default to using the 'select all' code of 00 (hex).

2.4 ITSO Shell detection

Detection of the presence of an ITSO Application (also known as the ITSO Shell) on a presented Media requires a POST to interrogate and parse certain data from the Media. The mechanism of this process is platform dependant. All POSTs shall provide such mechanisms for all platforms defined in ITSO TS 1000-10.

See 6.1 for the associated functional requirements.

2.5 Data entity support

All POSTs shall be capable of parsing and processing the ITSO data entities defined in ITSO TS 1000-2. These entities include, but are not limited to:

- the ITSO Shell Environment Data Group
- the Directory Data Group
- the IPE Data Group
- the Value Record Data Group
- the Cyclic Log
- the data structures that make up Data Groups
  - the Label
  - the Dataset
  - the Instance ID
  - the Seal

ITSO TS 1000-10 defines the data entity locations and access mechanisms for each platform type.

2.6 IPE support

The complete set of IPEs is defined in ITSO TS 1000-5.

A POST shall support the required set of IPEs that are relevant to its use within an interoperable environment. This required IPE set is defined by ITSO.

3. POST to HOPS interface

3.1 General

As described in ITSO TS 1000-9, the ITSO communications architecture is hierarchical, with POSTs at the 'base' of the hierarchy. At the 'apex' of each ITSO Compliant Scheme is a first line HOPS, as detailed in ITSO TS 1000-9 which forms the interface between the scheme and the rest of the ITSO Environment. Between the POST and the HOPS there are typically scheme specific layers, where data concatenation (for upward flows) and distribution (for downward flows) take place.

In order to provide the required data exchanges within the environment, all POSTs shall have the ability and means to carry out 2-way communications with a defined 'first-line' HOPS.

Secure, lossless data communication is an essential requirement of the ITSO Environment. Within a POST this is achieved by use of data retention rules, together with services provided by the ISAM for Data Frame sealing. Generation of a valid ITSO Message and verification of an ITSO Message is always a joint task involving the POST and its ISAM. For details of the ISAM functionality refer to ITSO TS 1000-7 and ITSO TS 1000-8.

Within the following clauses, the term POST is used to collectively describe the POST / ISAM ensemble for message generation and verification.

3.2 Mandated message set

Table 1 provides a summary of the messages that shall be supported by all POSTs. These are grouped by Message Class. (Rx = shall support reception of message; Tx = shall support transmission of message).

Table 1 - Message set summary

Message Class / Message	Requirement
Class 0	
ACK1 (Transmission Control)	Rx
ACK2 (Transmission Control)	Rx / Tx
NAK1 (Transmission Control)	Rx
NAK2 (Transmission Control)	Rx / Tx
Class 1	
Transaction Record Data	Tx
Class 2	
Query	Tx
Query Response	Rx
Configuration Data List	Rx
Parameter Table	Rx
Miscellaneous Messages	Rx / Tx
User Defined	Rx / Tx

Class 3	
ISAM Security File	Rx
ISAM Security Acknowledgement	Tx

ITSO provides a Class 2 User Defined message type that can be used between the POST and HOPS. Support for such a message type is not mandated.

The mandatory conditions that apply to a miscellaneous message are defined in TS1000–6 for each miscellaneous message code.

ITSO TS 1000-9 defines the generic structure that shall be used for ITSO Messages.

ITSO TS 1000-6 defines the content and usage rules that shall be followed for each message type.

3.3 Transmission methods and data formats

As defined in ITSO TS 1000-9, there is no mandated transmission method or data format for communications between a POST and its first-line HOPS. System designers may use any transmission method or data format, provided that the following requirements as detailed in ITSO TS 1000-9 are met:

- the methods and formats support the Loss Less data transmission methodology
- the methods and formats allow the required data to be transmitted in a robust and secure manner
- the methods and formats allow the transmitted data to be fully recovered to its native format
- the methods and formats allow the transmitted data to be verified against the ISAM generated Seal at the receiving node

It is strongly recommended that POSTs be able to transmit and accept data in the ITSO transmission format as defined in ITSO TS 1000-9.

If a POST does not support either the full or the minimal XML tag set, then its certification will be endorsed. This endorsement will state what other systems and/or utilities (e.g. depot systems, data converters, etc.) are required to enable an XML ITSO interface to be provided. In effect this endorsement will list what is required to interface the POST to a HOPS that provides an ITSO-defined XML data portal.

3.4 Loss Less data transmission

As defined in ITSO TS 1000-9, the ITSO communications environment provides end-to-end Loss Less data transmission at application level by the use of:

- Positive Acknowledgements
- Data retention & re-transmission

3.4.1 Positive acknowledgements

All POSTs shall generate and transmit the appropriate Class 0 Transmission Control message (ACK2 or NAK2) on receipt of a Class 2 message from a HOPS. Refer to 6.3 for the functional requirements relating to the generation of acknowledgements.

All POSTs shall receive and correctly process Class 0 Transmission Control messages received from a HOPS. Refer to 6.3 for the functional requirements relating to the processing of acknowledgements.

3.4.2 Data retention

Data retention by POSTs is an integral part of the ITSO Loss Less data transmission mechanism. All POSTs shall retain a full copy of data transmitted within Class 1 and Class 2 messages in a secure and non-volatile manner until a valid ACK is received for such data transmission. Then, and only then, shall a POST be at liberty to clear the data from that secure storage. See 6.3 for functional requirements relating to the processing of acknowledgements and determination of ACK validity.

Note: ITSO does not mandate that transmitted data be cleared from secure storage after successful transmission and acknowledgement, and POSTs are allowed to maintain a copy of such data.

Continued ITSO-specific operation of a POST shall be subservient to the above requirement for data retention. This means that if the memory store of a POST becomes 'full' and the POST cannot successfully transmit the data onward and receive valid positive acknowledgements of this data, then the POST shall immediately suspend normal ITSO operation until successful data transfer and acknowledgement has been restored. The POST shall signal 'out of service' as defined in clause 5.

Note: The above requirement will be a factor in determining the appropriate memory sizing and communications technology for use in a given system. The ISAM can be configured via the AMS system to store Transaction Records, thus providing the POST with additional secure, non-volatile memory resources. See 6.3 for further details.

3.4.3 Re-transmission of data

If a POST does not receive the required ACK for a previously transmitted message within the defined timeout period, then it shall re-transmit the message to the HOPS.

Note: ITSO TS 1000-9 only specifies the maximum allowable timeout period. A scheme may use a shorter timeout if its communications infrastructure supports this.

3.4.4 Removable memory modules

Certain designs or installations of POSTs may employ removable memory modules to provide data transfer between the POST and a fixed infrastructure (and eventually the 'first-line' HOPS). In these cases the following rules shall apply:

- The POST shall store Transaction Records (or use the ISAM to store them) and retain them in secure non-volatile storage within its fixed memory regardless of whether they are simultaneously written to the removable memory module

- The removable memory modules themselves are not subject to ITSO data retention and deletion rules, and are considered to be a form of transport medium between POST and HOPS

- POSTs may employ either of the following modes to store Transaction Records

  ○ update the memory module as Transaction Records are created in real time

  ○ write the Transaction Records to the memory module as a 'batch' operation

- If the Transaction Record storage in the POST (or ISAM if used) becomes full the POST shall cease ITSO-related operation as defined in 3.4.2. This applies regardless of whether or not the removable memory module has remaining storage capacity

- If the POST is operating in the first mode, i.e. that of updating the memory module continuously, and the module's memory becomes full while the POST (or ISAM if used) has remaining memory capacity, then ITSO does not require that the POST suspend ITSO related operation

4. POST to ISAM interface

4.1 General

The ISAM is an ITSO-supplied security sub-system, and is one of the main technical security components within the ITSO Environment. Refer to ITSO TS 1000-7 for further details of role and scope of the ISAM security sub-system.

Every instance of a POST operating within the ITSO Environment shall have an ISAM fitted.

4.2 Physical

The socket design shall allow the ISAM to be inserted and removed without risk of damage to the ISAM or POST. When 'closed' the socket shall lock the ISAM securely in place.

The socket shall be located such that tool access is required to access it. However, such access shall not require any tamper-proof or warranty seals to be broken.

For HOPS/AMS, the ISAM may remain mounted in its carrier and be inserted into a standard ITSO-compliant contact card reader or as appropriate.

The electrical contacts to the ISAM are as defined in [ISO/IEC 7816-2].

4.2.1 Legacy ISAM (H3 or earlier)

If the POST/HOPS is expected to support legacy ISAMs (H3 or earlier), it must accommodate an ID-1 or ID-000 format socket. This does not preclude the use of V4 ISAMs in these formats.

4.2.2 V4

The POST/HOPS may optionally support any of the following form factors: ID-1, ID-000, 3FF or 4FF (commonly known as Card, Micro, Mini and Nano SIM).

4.3 Electrical

4.3.1 device

- all POST/HOPS shall adopt parameters indicated within the ATR delivered by the ISAM. In particular, the POST/HOPS must only supply an indicated voltage.

- the POST shall be able to supply a continuous current of at least 50mA to the ISAM

- the POST shall use a clocking frequency to the ISAM in the range of 1MHz - 5MHz

- the POST shall be capable of supporting an I/O baud rate of 115Kbitss^{-1} at 3.579MHz. It is recommended that POSTs be capable of supporting the maximum ISAM I/O baud rate of 446.2Kbitss^{-1} at 3.579MHz

- all other electrical parameters shall be as defined in [ISO 7816]

4.4 Protocol

The POST shall communicate with the ISAM using the T=1 protocol as defined in ISO/IEC 7816-3:1997.

The POST shall act in the role of 'interface device' as defined by ISO/IEC 7816-3:1997.

Refer to ITSO TS 1000-8 for details of the Answer to Reset (ATR) response that the ISAM provides.

Refer to ITSO TS 1000-8 for details of the Protocol & Parameter Selection (PPS) settings supported by the ISAM.

4.5 Command set

Refer to ITSO TS 1000-8 for details of the full command set supported by the ISAM.

Refer to ITSO TS 1000-7 for the ISAM command sequences that shall be generated by the POST to carry out the operational functions required.

4.6 Usage of the ISAM

The POST shall use the services provided by the ISAM when carrying out the following functions:

- verification of ISAM validity at POST startup / initialisation
- mutual authentication between Customer Media and POST (where applicable)
- setting up a POST to Customer Media communications session
- closing down a POST to Customer Media communications session
- getting access keys for the ITSO Shell on presented Media
- getting access keys for the ITSO Directory on presented Media
- getting access keys for IPEs on presented Media
- creating an instance of an IPE on presented Media
- deleting an instance of an IPE on presented Media
- verification of the validity of the ITSO Directory on presented Media
- verification of the validity of IPEs within an ITSO Shell on presented Media
- verification of the validity of Value Records within an ITSO Shell on presented Media
- sealing of IPEs which have been modified by the POST
- sealing of Value Records which have been modified by the POST
- sealing of the ITSO Directory after modification by the POST
- sealing of Transaction Records generated by the POST
- initialising the IBatch Header
- computation of a running IBatch Header for Transaction Records generated by the POST
- clearing down an IBatch Header when the associated Transaction Records have all been successfully transferred to a HOPS
- verification of the validity of Data Frames received by the POST
- sealing of Data Frames generated by the POST
- searching of lists (hot / action) held within the ISAM

4.7 Secure Data Frame data handling

The POST shall pass any ISAM Security File received in a Secure Data Frame to the ISAM without delay. If a transaction is in progress, then this transaction may be completed before the Security File transfer is effected.

The POST shall retrieve the ISAM Security Acknowledgement from the ISAM using the mechanism defined in ITSO TS 1000-8 and place this Secure Data Frame into a Class 3 message and transmit this message to the HOPS as soon as is possible.

4.8 Exception handling

The ISAM returns a status code in response to commands from the POST (see ITSO TS 1000-8). This status code is classified into one of the following categories:

• OK	Command executed successfully
• NON-FATAL ERROR	An error occurred but POST operation may continue
• FATAL ERROR	An error occurred and the POST must terminate ITSO operation

POSTs shall check the status code after each command and based on the classification of this code shall take the appropriate action as defined above.

It is strongly recommended that the POST log the detailed error code to allow fault diagnostics.

5. Human interface

5.1 General

Specification of certain attributes of the human interface by ITSO are required to ensure that customers receive a consistent 'user experience' when faced with an ITSO-enabled POST. These ITSO-specific attributes form only a part of the overall human interface of the unit, and as previously stated, other essential scheme-specific requirements will need to be specified for realising an operational system.

Card terminals used by members of the public should comply with:

- accessibility requirements of the Equality Act 2010
- [EN 1332-1] - Identification card systems; man-machine interface - Design principles for the user interface
- [EN 1332-3] - Identification card systems; man-machine interface - Keypads

5.2 Accessibility

The Media Target Area shall be clearly indicated to customers.

Accessibility to the Media Target Area shall conform to the requirements of the Equality Act 2020.

Accessibility to the Media Target Area shall conform to the requirements of [EN 1332-1].

Accessibility to any keypads required by the customer shall conform to the requirements of the Equality Act 2010.

Accessibility to any keypads required by the customer shall conform to the requirements of [EN 1332-3].

5.3 Customer signals

POSTs used in applications where the customer interacts directly with the unit shall provide both audible and visual signals to assist usage of Media carrying an ITSO Shell.

5.3.1 Audible signals

Audible signals should conform to the requirements of [EN 1332-1].

Audible signals shall be used to indicate and differentiate:

- completion of a successful ITSO Transaction
- an unsuccessful ITSO Transaction

5.3.2 Visual signals

Visual signals should conform to the requirements of [EN 1332-1].

Visual signals shall be used to indicate and differentiate:

- a POST that is 'In Service' and able to accept and process Media carrying an ITSO Shell
- a POST that is 'Out of Service' or unable to accept and process Media carrying an ITSO Shell
- completion of a successful ITSO Transaction
- an unsuccessful ITSO Transaction

5.4 Other I/O devices

5.4.1 Customer display

In addition to the visual signals defined in section 5.3.2, POSTs may have a display for conveying more detailed information to a customer.

Such a user display, if present, should conform to the requirements of [EN 1332-1].

If Media is presented that carries 'User Related Information' (URI) coding, then the POST should adapt any customer display settings to conform to the requirements of [EN 1332-4].

The message strings associated with ITSO Transactions are defined in Annex A – Standard customer Messages.

5.4.2 User console

POSTs may have a user console for allowing a customer to input and view data associated with their Customer Media and / or Transactions.

Such a user console, if present, should conform to the requirements of [EN 1332-1] and [EN 1332-3].

If Media is presented that carries 'User Related Information' (URI) coding, then the POST should adapt any user console settings to conform to the requirements of [EN 1332-4].

5.5 ITSO accreditation marking

Instances of a POST design that has been certified as ITSO Accredited, and are used in a scheme where ITSO Customer Media is accepted shall bear the approved ITSO Accreditation logo. This logo shall be situated in a location that is visible without requiring the POST to be removed from any mounting.

Refer to the ITSO Business Rules for further details.

6. Functional requirements of the POST software

This clause defines the functional requirements relating to:

- media handling
- IPE handling
- message generation and processing
- configuration handling

6.1 Media handling

This sub-clause defines the following functional requirements relating to handling of ITSO Customer Media:

- detection and validation of the ITSO Shell
- validation of the Directory
- selection of products
- handling of Anti-tear
- media re-presentation
- transaction time

6.1.1 Detection and validation of the ITSO Shell

As stated in clause 2, all POSTs shall be able to automatically detect and initiate communications with all Customer Media platforms (CMDs) defined in ITSO TS 1000-10.

Because of the range of platform types, there is no common mechanism that can be used to establish the existence or otherwise of an ITSO Shell on presented Media. The POST shall, for each CMD, carry out the required interrogation of the Media to establish if an ITSO Shell is present or not. See ITSO TS 1000-10 for details of ITSO Shell detection.

At the end of the above process, if the presented Media contains a valid, non-expired ITSO Shell, then the POST will have the required parameters to access and validate the Directory. These parameters include:

- MID
- ISRN
- FVC
- KSC
- KVC
- KAS   (Value determined by data entity being accessed)

Refer to ITSO TS 1000-2 and ITSO TS 1000-10 for further details on the meaning and usage of these parameters.

If the POST could not detect an ITSO Shell, or the detected ITSO Shell has expired, then the POST shall terminate the transaction and shall record an exception Transaction Record as defined in ITSO TS 1000-6.

6.1.1.1 Compact ITSO Shell processing

The concept of a 'Compact Shell' is defined in ITSO TS 1000-2. This form of ITSO Shell is used on platforms with limited storage capacity. See ITSO TS 1000-10 for details of such platforms.

When a POST detects a platform carrying a Compact ITSO Shell, then it shall 'reconstitute' a full ITSO Shell using the parameters defined in ITSO TS 1000-10. It shall use this 'reconstituted' ITSO Shell where required (e.g. for sending parameters as part of ISAM commands).

6.1.2 Validation of the Directory

ITSO Shell detection as described in the above sub-clause does not require the services of the ISAM. However, all further interaction with the platform must take place within a 'secured session'. This session requires the POST to make use of a number of ISAM services during dialogue with the Media. ITSO TS 1000-7 defines the detailed structure of a secured session.

When a valid ITSO Shell has been detected, the POST shall commence and execute a secured session as defined in ITSO TS 1000-7 and ITSO TS 1000-8.

The ISAM will signal an error condition if it determines that:

- the Format Version Code is not supported or authorised for use on the POST

- the Key Strategy Code is not supported or authorised for use on the POST

In both of the above cases the POST shall terminate the transaction and the session, and shall record an exception Transaction Record as defined in ITSO TS 1000-6.

The next phase of the secure session is Directory validation. During this phase the POST obtains the required keys from the ISAM to access the Directory and then passes the contents of that Directory to the ISAM for verification of the Directory Seal.

On platforms that have Anti-tear copies of the Directory, then checking of the copies shall be done in the manner defined in ITSO TS 1000-10.

If the ISAM indicates that the ITSO Shell contains a valid Directory, then the POST shall continue the transaction by establishing that the ITSO Shell is not blocked or Hotlisted.

If the ISAM indicates that the ITSO Shell contains an invalid Directory, then the POST shall continue the normal rules of tear handling and shall record an exception Transaction Record – Exception Type code 5 as defined in ITSO TS 1000-6.

If a valid Directory is not found the POST shall terminate the transaction and the session, and shall record an exception Transaction Record - Exception Type code 5 as defined in ITSO TS 1000-6.

If the ITSO Shell is Hotlisted or blocked the POST shall terminate the transaction and the session, and shall record an exception Transaction Record - Exception Type code 134 as defined in ITSO TS 1000-6.

If the Media contains a valid Directory, and its ITSO Shell is not blocked or Hotlisted then the POST shall continue the transaction by establishing the presence of suitable products(s) within the ITSO Shell.

6.1.3 Selection of products

The POST shall parse the Directory Entries to establish the presence (or otherwise) of products that are suitable for the application in which the POST is operating. The POST can check for Entries that:

- contain an IIN, OID, TYP, PTYP combination that the POST has operational / Interoperability rules for;

- are not blocked for use;

- are not expired, with due regard to ExpiryTime, which may indicate that the product remains valid until a specified time on the day following the last day of validity.

Following this initial parsing, the POST shall assemble a 'candidate list' of products (IPE instances) that can then be verified. Within this context, IPE instance shall refer to both the 'static' IPE Data Group and any associated Value Record Data Group.

If no candidate IPE instances are found the POST shall terminate the transaction and the session, and shall record an exception Transaction Record as defined in ITSO TS 1000-6.

The selection order for candidate IPE instances shall be made using:

- configured selection rules[1] held within the POST, where such rules are agreed between the Service Operator and the Product Owner(s)

- IPE PriorityOverride flag settings

- selection by the customer if the operating environment and POST design permits this

The POST shall then normally submit the selected IPE instance to the ISAM for verification of validity in the manner defined in ITSO TS 1000-7 and ITSO TS 1000-8. If the POST is performing an entry transaction into a check in / check out system (where product selection is performed on exit) then this stage may be optionally omitted. It should however be noted that the risk of the use of fraudulent products is increased by not checking the individual product seals.

If the IPE instance is deemed to be invalid for use by the ISAM the POST shall record an exception Transaction Record as defined in ITSO TS 1000-6. If there is an alternative IPE instance, then this shall be submitted to the ISAM for verification.

If a valid IPE instance cannot be found, then the POST shall terminate the transaction and the session, and shall record an exception Transaction Record as defined in ITSO TS 1000-6.

If a valid IPE instance is found then the POST shall process the IPE as defined in 6.2.

6.1.4 Handling of Anti-tear

All POSTs shall implement and support the Anti-tear mechanisms outlined in ITSO TS 1000-2 and defined in ITSO TS 1000-10.

A POST shall not transact with Customer Media in a manner that will render the Media unreadable by other POSTs.

When a POST detects premature removal of Media, it shall signal an error condition to the user, and wherever possible, prompt for the Media to be re-presented. The POST shall make a record of the status of the transaction, such that if the Media is re-presented, then the transaction can be completed.

A POST shall always complete a transaction if the Customer Media in question is re-presented before any other Media is presented. It is recommended that POSTs be able to complete the transaction on Torn Customer Media if it presented within a defined number of transactions (say 10).

If a POST is unable to verify that the Media was correctly updated, it shall not create a 'successful' Transaction Record. Instead it shall record an exception Transaction Record as defined in ITSO TS 1000-6.

6.1.5 Media re-presentation

POSTs shall be designed to prevent 'Double Transactions' due to inadvertent re-presentation of Customer Media. The mechanisms used shall take into consideration the application, the operating environment and the ticketing products used.

Note: Certain applications may require the system to legitimately carry out relatively rapid transactions on a single platform; group use of a carnet product or Stored Travel Rights (STR) being two examples.

6.1.6 Transaction time

6.1.6.1 Benchmark Transaction

POSTs shall be able to carry out the defined benchmark Transactions defined in ITSO TS 1000-10.

---

[1] These rules shall aim to ensure that the customer is given 'best value' based on the available options.

The benchmark Transaction shall be deemed to start when the Media has fully entered the RF field of the POST.

The benchmark Transaction shall be deemed to end when the POST has received the last data item required from the Media as defined in ITSO TS 1000-10.

The benchmark Transaction shall not require any operator or customer input.

6.1.6.2 Throughput

POSTs shall be capable of maintaining an average sustained throughput of 3 transactions per second for the benchmark transaction described in the above sub-clause.

The maximum dwell time between cessation of Media activity from a transaction and being ready to commence activity with a subsequent user shall be no more than 100 ms.

6.2 IPE handling

This sub-clause defines functional requirements relating to handling of IPE instances present on the Customer Media. As before, unless otherwise stated the term IPE instance shall refer to both the 'static' IPE Data Group and any associated Value Record Data Group.

The following areas are covered:

- general IPE instance processing
- IPE instance creation
- IPE instance deletion
- Cyclic Log updating
- Auto-Renew
- Stored Travel Rights (STR) processing
- Ticket TransactionReversal
- printing of Tickets and receipts

6.2.1 General IPE instance processing

An IPE instance is an instantiation of an IPE embodiment specification. This embodiment specification defines, for a given IIN, OID, TYP and PTYP, the allowable ranges and usages of the data elements contained within the IPE. In addition, the embodiment specification defines the commercial purpose and mode of usage of the product (e.g. a weekly travelcard valid in a given area).

The ISAM within the POST enforces agreed business relationship rules, and determines if the product defined by IIN, OID, TYP and PTYP is valid for use at the POST in question (i.e. does the Product Owner of the product have an ITSO-registered agreement with the Service Operator of the POST).

The ISAM does **not** hold or control rules relating to the embodiment, which are defined by the Product Owner. The POST shall be configured with all such rules, and shall use those rules in a complete and accurate manner.

In addition to these product-specific embodiment rules, the POST shall always perform the following during the processing of an IPE instance:

- Apply any pending actions that are required, based on data held in the Actionlist. Those actions shall be applied in the sequence defined by their associated sequence number
- Verify temporal validity of the product. This shall include 'Anti-passback' checks if relevant (see 6.2.1.1)
- Use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to generate updated Seals for the IPE Data Group and/or the Value Record Data Group of the IPE instance if changes are made to data elements with those groups

- Update the Cyclic Log if this is present

- Generate an updated copy of the Directory with the new entry and use the services of the ISAM to seal that directory. The POST shall substitute the value of KID in the Directory Instance Identifier presented for re-sealing with the value that is held in the POST if this is greater (allowing for rollover) than the value of KID used when verifying the directory seal. Details of the latest key version to use may be read from the ISAM

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

- Correctly implement rollover on data fields within IPEs in the manner defined in ITSO TS 1000-5

6.2.1.1 Anti-passback

'Anti-passback' is a period of time during which a Product may not be presented more than once, and is designed to prevent the deliberate multiple use of a travel permit by multiple users.

The POST shall support Anti-passback checking where this is required by a product. If Media carrying a Product that uses Anti-passback is presented twice within the number of minutes defined in the Product's 'PassbackTime' field, the POST shall signal an error, and shall not process the Product; however the ITSO Shell and other Products within the ITSO Shell are still valid for use.

The passback time appropriate to a product is stored in the IPE in the PassbackTime data element.

- If there is no PassbackTime data element within an IPE type, then application of antipassback is optional

- A value of zero in the PassbackTime data element shall indicate that passback time is not defined within the IPE, in which case anti passback rules defined within the POST shall be implemented.

Where the IPE PassbackTime data element contains zero, then there are three modes for passback time rules defined in the POST, all of which may be required simultaneously:
1. Passback is not required
2. Passback time is defined as a parameter within the POST
3. Passback time has been defined centrally and disseminated to the POST in a Passback Times PCD message. Note that the applicable Anti-passback rule may depend upon the type of IPE being processed, i.e. a POST should be capable of supporting all the above options. Where option 2 applies, then care shall be taken that the Anti passback function is designed in such a way that it is applied correctly for all the types of IPE which will be encountered, including those Products where multiple presentation within a short space of time is allowed.

Note: A DTS, passback time and IPE reference relating to the last relevant IPE used is stored in the Log directory entry. This data may be used to quickly determine if an IPE is invalid due to Anti-passback rules. If two IPEs from the same CM have been used within the passback period, then either the second copy of the directory where present or any Transient Ticket data may be used to determine whether the second is still within its passback period.

6.2.2 IPE instance creation

The POST shall use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to establish if it has authority to create an IPE instance of a given IIN, OID, TYP and PTYP.

If the POST is authorised to create the IPE instance, then it shall carry out the following actions:

- Verify that there is a 'free' Directory Entry available. If a free entry is not available, and there are no expired IPE instances available for deletion (see 6.2.3), then the IPE instance shall not be created

- Verify that there are sufficient storage sectors free to hold the IPE instance (IPE Data Group and any associated Value Record Data Group). If free sectors are not available, and there are no expired IPE instances available for deletion (see 6.2.3), then the IPE instance shall not be created

- Create the required IPE Data Group and use the services of the ISAM to seal this

- Create the associated Value Record Data Group (if required) and use the services of the ISAM to seal this

- Write the above data group(s) to the Media

- Generate an updated copy of the Directory with the new entry and use the services of the ISAM to seal that directory

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

It should be noted that because ExpiryTime values may exceed 1440 (midnight), then products may remain valid on the day following the last day of validity. To avoid automatic removal of products on the day following the day specified by EXP, Product Owners shall configure RemoveDate with a value greater than zero when ExpiryTime contains a value greater than 1439.

6.2.3 IPE deletion

Where there is insufficient space in the ITSO Shell to create an IPE instance, a POST may be authorised to delete expired IPEs created by other Product Owners.

The POST shall initially verify that both the Expiry Date in the Directory Entry and the date defined by RemoveDate in the IPE instance have passed. Note that:

- if the Expiry Date is set to zero, then the POST shall not delete the IPE under any circumstances

- if the RemoveDate field is set to 255, then the IPE may not be deleted

- If both Expiry Date and RemoveDate are such that deletion is allowed, then the POST shall use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to establish if it has authority to delete the IPE instance

- If the POST is authorised to delete the IPE instance, then it shall carry out the following actions

- Create the required (replacement) IPE Data Group and use the services of the ISAM to seal this

- Create the associated Value Record Data Group (if required) and use the services of the ISAM to seal this

- Write the above data group(s) to the Media

- Generate an updated copy of the Directory with the new entry and use the services of the ISAM to seal that directory

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

6.2.4 Cyclic Log updating

The POST shall check if a Cyclic Log is present in the ITSO Shell of the Media transacted with. The Log is fully defined in ITSO TS 1000-2.

6.2.4.1 Basic Mode Log

A basic mode Log change is conducted when use of an IPE is being recorded, but a Cyclic Log entry is not being written.

When a basic mode Log change is made then the POST shall update the following data elements of the Log Directory Entry. This update shall take place after any updates of the IPE Data Group(s) but prior to updates (and sealing) of the Directory.

• PTR	Updated to the Directory Entry index of the most relevant IPE used in the transaction
• EEI	Updated to the required 'nesting level' if the transaction is associated with a closed system
• DTS	Updated with date and time of transaction
• PTLBM	Updated with the Anti-passback time appropriate to the IPE pointed to by PTR
• LPF	Set this flag to indicate a basic mode entry

Other elements of the Log Directory Entry shall not be changed.

6.2.4.2 Normal Mode Log

A normal mode Cyclic Log change shall only be made when the conditions for adding a Transient Ticket Data Record specified in ITSO TS 1000-5 are satisfied. The data set stored in the Cyclic Log shall conform to the definition of a Transient Ticket Data Record as specified in ITSO TS 1000-5. The directory entry shall be updated as defined above for basic mode, excepting that the LPF flag shall be set so as to indicate a normal mode entry, and the record offset RO shall be set to indicate the next record to be written.

Refer to ITSO TS 1000-6 for definitions of the Transaction Record type.

6.2.5 Auto-Renew

The POST shall check if the product to be used supports Auto-Renew (as defined in ITSO TS 1000-5), and shall have the capability to carry out this operation if required.

If the IPE instance has its Auto-Renew flag set, and the entitlement available has run out / expired, then the POST shall use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to establish if it has authority to 'top-up' the IPE instance.

If the POST is authorised to Auto-Renew the IPE instance, then it shall carry out the following actions:

- Modify the associated Value Record Data Group as required and use the services of the ISAM to seal this

- Generate an updated copy of the Directory with the new entry and use the services of the ISAM to seal that directory

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

6.2.6 Stored Travel Rights processing

6.2.6.1 STR presence

ITSO does not mandate all Media platforms used in ITSO Compliant Schemes contain a National STR IPE instance.

If the ITSO Shell Owner wishes to provide a local STR scheme, then they shall create one TYP 2 IPE instance.

6.2.6.2 STR ownership transfer

Providing that the STR product on a platform has not been used then POSTs shall have the capability to change the owning OID of that product.

POSTs shall have the capability to change the owning OID of the STR product. Such a change shall only be allowed if:

- The STR product instance on the Media has not been used (i.e. the coding in the Sector Chain Table indicates that the IPE has never been used as defined in ITSO TS 1000-2) and

- The services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) were used to establish if the POST has the authority to carry out such a change. Such authority can be identified where the ISAM contains the appropriate keys and permissions for both the old and new OIDs

If the POST is authorised to change the owning OID, then it shall carry out the following actions atomically:

- Delete the existing IPE Data Group and the associated Value Record Data Group and generate the appropriate code 0007 Transaction Record message. Where the IPE owner was identified by OID 8000 then the message shall be held by the POST owner's HOPS, and not sent to the IPE owner, pending instructions from ITSO on its disposition

- Create the new IPE Data Group and associated Value Record Data Group and use the services of the ISAM to seal this

- Generate an updated copy of the Directory with the new entry and use the services of the ISAM to seal that directory

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state. The following message types shall normally be generated: code 0005 "Create IPE" sent to the Shell owner; and code 0120 "Create Stored Travel Rights TYP 2" sent to the POST and IPE owners

6.2.6.3 STR load limits

The ISAM is designed to control the STR loading / incrementing. The checks and controls are:

- The ability, or not, to increment the STR product on any Customer Media

- The maximum value that can be held in the STR product on a given instance of Customer Media

- The maximum aggregate value that can be added by the POST (over multiple Media instances)

POSTs shall always use the services of the ISAM to ensure that these checks and controls are adhered to.

6.2.6.4 STR loading

As defined in the above sub-clause, STR loading / incrementing by a POST is subject to checks and controls by the ISAM. POSTs shall use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to carry out such loading / incrementing in the approved manner.

If the POST is authorised to load the STR product, then it shall carry out the following actions:

- Modify the associated Value Record Data Group and use the services of the ISAM to seal this

- Generate an updated copy of the Directory with the new entry and use the services of the ISAM to seal that directory

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state.

6.2.6.5 Reporting of the addition of STR value

If the STR product is used in a Transaction, the POSTs shall check for the presence of an 'add value' record in the STR IPE Value Record Data Group.

If such a record is found, the POST shall use it to create a Transaction Record (of message code 0103) pertaining to this previous 'add value' transaction. This Transaction Record shall be transmitted along with all the normal Transaction records for the actual transaction carried out.

Note: This mechanism offers a measure of protection in case the original 'add value' Transaction Record was not transmitted or received.

6.2.6.6 STR priority

The STR IPE contains a priority override flag. The definition of this flag is that it indicates to a POST to use the STR product in preference to any other payment mechanism contained within the Customer Media.

Because STR product selection is determined by external rules applied to the candidate product list (see 6.1.3), such rules have to also provide priority to the STR product in order for the product to be checked and parsed and this flag read by the POST.

6.2.6.7 Auto-Top-Up

Auto-Top-Up enables a STR IPE instance to be automatically 'topped-up' to a preset value as and when a defined minimum threshold is reached. See ITSO TS 1000-5 for the associated thresholds.

The POST shall check the Auto-Top-Up start date to ensure that Auto-Top-Up is permitted at this time even though the flag is set to indicate Auto-Top-Up is allowed. This date is used for newly created STR product instances to allow time for customer credit worthiness to be checked.

The POST shall only process one Auto-Top-Up action per transaction.

In the case of Auto-Top-Up flagged as being supported from an external purse, the POST shall indicate this to the operator and shall terminate the ITSO Auto-Top-Up sequence. As and when (external) top-up has taken place, the ITSO Transaction may be re-attempted.

6.2.6.8 STR First Use

A first use detection facility is provided which may be used by the Product Owner for marketing and or security purposes. The rules which determine when a first use event takes place shall be determined by the Product Owner

and programmed into POSTs (including Personalisation POSTs), and may include any one of the following example options:

- any Transaction taking place or

- any Transaction instigated by the card holder taking place or

- a Transaction instigated by the card holder where Stored Value is used to make a purchase or

- a Transaction instigated by the card holder to add value to Stored Value

When a first use event is detected then the POST shall:

- set the directory data group sector chain table array to indicate used as defined in ITSO TS1000-2

- send a 0008 message to the Shell Owner; and

- send a 0106 message to the POST and Product Owners

6.2.7 TransactionReversal

If the POST supports Product TransactionReversal and the capability criteria tables in the ISAM are set to allow IPE deletion, then it shall reverse the Transactions which created Products stored on the Media as follows:

- Delete the IPE using the Delete IPE command

- Generate an updated copy of the Directory, with the IPE entry deleted and use the services of the ISAM to seal the directory

- Write the updated Directory to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

Notes:

Reversal of a TransactionReversal is achieved by recreating the IPE. This can be carried out in the same ISAM session as the IPE deletion described above. The process is the same as for IPE instance creation (clause 6.2.2).

Where the platform utilises a one-time programmable area for the usage of a product then TransactionReversal will not be possible.

If a Transient Ticket record was created in the Cyclic Log it shall be cancelled by overwriting. If no valid Transient Ticket data is required then write null data.

Any stored rides deducted from a Ticket IPE shall be restored.

If Stored Travel Rights or Charge To Account was used as a method of payment in the original Transaction, then the account shall be restored to its previous value.

6.2.8 Printing of Tickets and receipts

If the POST has a printer, then it shall issue a Ticket or receipt as defined by the control flags within the IPE instance used (see ITSO TS 1000-5).

6.2.9 Creation of IPE Value Groups containing Value

The actions of creating an IPE with a Value Group, and of adding value to that IPE, shall be treated as two separate Transactions, and all the relevant Transaction Record messages created. However a single card write activity is permissible.

In the case where no value is added to the IPE upon creation, then all TransactionSequenceNumber elements shall be set to zero.

In the case where value is added to the IPE at the time of creation, then the TransactionSequenceNumber element in one copy of the Value Group shall be incremented, reflecting the second transaction which added value to the IPE. In this case the TransactionSequenceNumber embodiment specification data element shall either be "set to value in embodiment spec or set to value determined upon IPE creation".

6.3 Message generation and processing

As defined in clause 3, all POSTs shall support 2-way communication to a HOPS. Four classes of ITSO application message, as defined in ITSO TS 1000-9, are exchanged between the POST and the HOPS:

- Class 0   Positive acknowledgement messages
- Class 1   Batch-oriented POST to HOPS message
- Class 2   General frame-oriented message
- Class 3   ISAM security message

All POSTs shall support the above message classes and shall generate and process those messages as follows:

6.3.1 Class 0 message generation

Each Data Frame within a Class 0 message will be one of the following (see ITSO TS 1000-6 and ITSO TS 1000-9):

- ACK1
- ACK2
- NAK1
- NAK2

The only Data Frame types that the POST generates for Class 0 messages are ACK2 and NAK2. These Data Frames are generated in response to the POST receiving a Class 2 message.

6.3.1.1 ACK2 generation

ACK2 Data Frames are generated by the POST and sent to a HOPS when the POST receives a valid Class 2 Data Frame from the HOPS.

The POST shall determine the validity of the received Class 2 message and its constituent Data Frames as defined in ITSO TS 1000-9. This is summarised as follows:

- Verify the integrity of the overall message using the Message CRC
- Verify if authorised to process the message
- Verify the integrity of each Data Frame using the necessary ISAM services (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to carry out such verification
- For each correctly validated Data Frame, extract the data elements contained within the Data Frame and store / process these in the appropriate manner (see ITSO TS 1000-6)
- For each correctly validated Data Frame, extract the Data Frame sequence number parameter (see ITSO TS 1000-6)
- Using that parameter data, generate an ACK2 Data Frame and use the services of the ISAM to seal said Data Frame
- At the earliest possible time, transmit this ACK2 Data Frame to the HOPS as part of a Class 0 message

6.3.1.2 NAK2 generation

NAK2 Data Frames are generated by the POST and sent to a HOPS when the POST receives an invalid Class 2 Data Frame from the HOPS.

The POST shall determine the validity of the received Class 2 message and its constituent Data Frames as defined in ITSO TS 1000-9. This is summarised as follows:

- Verify the integrity of the overall message using the Message CRC

- Verify if authorised to process the message

- Verify the integrity of each Data Frame using the necessary ISAM services (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to carry out such verification

- For each Data Frame that was not correctly validated, extract the Data Frame sequence number parameter (see ITSO TS 1000-6)

- Using that parameter data, generate a NAK2 Data Frame and use the services of the ISAM to seal said Data Frame

- At the earliest possible time, transmit that NAK2 Data Frame to the HOPS as part of a Class 0 message

6.3.2 Class 0 message processing

The POST shall process received Class 0 messages as defined in ITSO TS 1000-9. This is summarised as follows:

- Verify the integrity of the overall message using the Message CRC

- Verify if authorised to process the message

- Verify the integrity of each Data Frame using the necessary ISAM services (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to carry out such verification

- Verify the destination of each Data Frame is correct (i.e. to the POST / ISAM in question)

Processing of the different types of Class 0 Data Frame is detailed in the following sub-clauses.

6.3.2.1 ACK1 processing

ACK1 Data Frames are generated by a HOPS and sent to a POST.

The purpose of the ACK1 is to explicitly acknowledge correct and full receipt (by the HOPS) of a batch of (Class 1) Transaction Records that were sent previously by the POST. This collection of Transaction Records is termed a Transaction Session Batch, and may have been transmitted between POST and HOPS over a period of time in multiple Class 1 messages (see ITSO TS 1000-9).

The POST shall process received ACK1 Data Frames as follows:

- Verify message integrity, Data Frame integrity and Data Frame destination as defined in 6.3.2

- For each correctly validated Data Frame, extract the IBatch Header sequence number and the IBatch Header delete parameters (see ITSO TS 1000-6)

- Using that parameter data, generate the required ISAM command (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to 'clear down' the IBatch Header stored within the ISAM for the acknowledged message batch

Note: ITSO does not mandate that the POST delete stored Transaction Records after the batch to which they are associated has been ACK'ed. The decision to retain or remove the Transaction Records is left to POST designers.

6.3.2.2 ACK2 processing

ACK2 Data Frames are generated by a HOPS and sent to a POST.

The purpose of the ACK2 is to explicitly acknowledge correct receipt (by the HOPS) of a Data Frame sent by the POST within a Class 2 message.

The POST shall process received ACK2 Data Frames as follows:

- Verify message integrity, Data Frame integrity and Data Frame destination as defined in 6.3.2

- For each correctly validated Data Frame, extract the Data Frame sequence number parameter (see ITSO TS 1000-6)

- Using that parameter data, determine the (Class 2) Data Frame to which the ACK2 pertains and mark said Data Frame as being successfully transmitted

Note: ITSO does not mandate that the POST delete stored Class 2 Data Frames after they have been ACK'ed. The decision to retain or remove the Data Frame is left to POST designers.

6.3.2.3 NAK1 processing

NAK1 Data Frames are generated by a HOPS and sent to a POST.

The purpose of the NAK1 is to explicitly indicate incorrect receipt (by the HOPS) of a Transaction Session Batch.

The POST shall process received NAK1 Data Frames as follows:

- Verify message integrity, NAK1 Data Frame integrity and NAK1 Data Frame destination as defined in 6.3.2

- For each correctly validated NAK1 Data Frame, extract the IBatch Header sequence number parameter (see ITSO TS 1000-6)

- At the earliest possible time, re-transmit all the Transaction Records that make up the Transaction Session Batch that is associated with that parameter data

6.3.2.4 NAK2 processing

NAK2 Data Frames are generated by a HOPS and sent to a POST.

The purpose of the NAK2 is to explicitly indicate incorrect receipt (by the HOPS) of a Data Frame sent (by the POST) within a Class 2 message.

The POST shall process received NAK2 Data Frames as follows:

- Verify message integrity, NAK2 Data Frame integrity and NAK2 Data Frame destination as defined in 6.3.2

- For each correctly validated NAK2 Data Frame, extract the Class 2 Message Data Frame sequence number parameter (see ITSO TS 1000-6)

- At the earliest possible time, re-transmit (as part of a Class 2 message) the Class 2 Message Data Frame that is associated with that parameter data

6.3.3 Class 1 message generation

Each Data Frame within a Class 1 message contains a Transaction Record (see ITSO TS 1000-6). Transaction Records form the basis of POST to HOPS data exchange.

All POSTs shall generate all required Transaction Records for activity carried out by the POST. ITSO TS 1000-5 and ITSO TS 1000-6 define what records shall be generated, and the data content and format of such records.

ITSO defines the concept of a Transaction Session Batch. This consists of a set of Transaction Records that all share a common IBatch Header (see ITSO TS 1000-9). The IBatch Header is automatically computed / updated by the ISAM when its services are used to seal a Transaction Record within a (Class 1) Data Frame.

All Transaction Records shall form part of a Transaction Session Batch. The POST shall use the services of the ISAM to initialise a Transaction Session Batch and generate an associated IBatch Header prior to carrying out transactions.

Note: ITSO does not prescribe the length of time or the number of transactions that a Transaction Session Batch may span. The only requirement is that there must be an 'open' Transaction Session Batch, with its associated IBatch Header, before the POST is allowed to carry out transactions.

Each Transaction Record within a Data Frame shall contain the data elements defined in ITSO TS 1000-6, and shall be sealed using the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8).

A Class 1 message shall comprise of one or more (sealed) Data Frames, together with the IBatch Header that the ISAM generated for the 'newest' Data Frame within the message. See ITSO TS 1000-7 and ITSO TS 1000-8 for details of how the IBatch Header is accessed.

The POST shall store a copy of all Data Frames transmitted as part of a Transaction Session Batch, together with the relevant IBatch Header, until an ACK1 has been received from the HOPS for said Transaction Session Batch. That storage shall be in a secure and non-volatile medium.

If the POST has 'closed' a Transaction Session Batch and indicated this to the HOPS, then it shall start an 'ACK timeout' as defined in ITSO TS 1000-9. If the required ACK1 is not received after this timeout expires, then the POST shall resend the Data Frames that make up said Transaction Session Batch to the HOPS.

Note: The ISAM can be configured via the AMS system to store Transaction Records, thus providing the POST with additional secure, non-volatile memory resources

6.3.3.1 Transaction Records destinations

The POST shall ensure that Transaction Records are generated and distributed to all required destinations, where those destinations are associated with the relevant entities in the ITSO Environment. These actors are:

- the ITSO Shell Owner
- the ITSO Shell Retailer
- the Product Owner
- the Product Retailer
- the Service Operator

ITSO TS 1000-6 defines the required distribution of each Transaction Record type.

ITSO TS 1000-9 defines how Transaction Record Data Frames can be given multiple destinations.

6.3.4 Class 1 message processing

Class 1 messages are only ever generated by a POST and sent to a HOPS. As such a POST shall never have to process such messages.

6.3.5 Class 2 message generation

Class 2 messages generated by a POST shall contain Data Frames with the correctly assigned 'POST to HOPS' message code (see ITSO TS 1000-6 and TS 1000-9).

A Class 2 message shall comprise of one or more (sealed) Data Frames. The POST shall use the services of the ISAM to seal those Data Frames.

The POST shall store a copy of each Data Frame transmitted, until an ACK2 has been received from the HOPS for said Data Frame. That storage shall be in a secure and non-volatile medium.

On sending Class 2 Data Frames, the POST shall start an 'ACK timeout' as defined in ITSO TS 1000-9. If the required ACK2 is not received after this timeout expires, then the POST shall resend the required Data Frame(s) to the HOPS.

6.3.6 Class 2 message processing

Class 2 messages received by a POST will consist of Data Frames, each of which will have one of the following message codes:

- Query Response

- Configuration List

- Parameter Table

- The POST shall process received Class 2 messages as defined in ITSO TS 1000-9. This is summarised as follows

- Verify the integrity of the overall message using the Message CRC

- Verify if authorised to process the message

- Verify the integrity of each Data Frame using the necessary ISAM services (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to carry out such verification

- For each correctly validated Data Frame, extract the Data Frame sequence number parameter (see ITSO TS 1000-6). Using this parameter data, generate an ACK2 Data Frame and seal this Data Frame using the services of the ISAM. See also 6.3.1.1

- For each Data Frame that was not correctly validated, extract the Data Frame sequence number parameter (see ITSO TS 1000-6). Using this parameter data, generate an NAK2 Data Frame and seal this Data Frame using the services of the ISAM. See also 6.3.1.2

- At the earliest possible time, transmit (as part of a Class 0 message) these Data Frames

- For each correctly validated Data Frame, extract the data elements contained within the Data Frame and store / process these in the appropriate manner (see ITSO TS 1000-6).

— Where a POST is in receipt of an ActionList message, there may be instances where the data frame is correctly validated, however the message contains conflicting data (for example, in a Create IPE action where the encoding for specific elements is contradictory). In such instances, the POST shall match the Shell on presentation, but not perform the action and return an 0312 unsuccessful match event with 0312ActionTaken set to 143.

6.3.7 Class 3 message generation

Class 3 messages generated by a POST shall contain Secure Data Frames generated by the ISAM, together with a POST-generated Batch Header.

After passing a received ISAM Secure Data Frame to the ISAM (see 6.3.8), the POST shall retrieve the ISAM Security Acknowledgement, which the ISAM will generate in the form of a Secure Data Frame (see ITSO TS 1000-7 and ITSO TS 1000-8.

All the Security Acknowledgments generated as the result of the application of the Secure Data Frames received by the POST within a single Class 3 message including those found to have already been applied (see clause 6.3.8.1.2 APPLY FRAME function) shall be packed together in a single Class 3 message in the same order as they were generated. The POST shall then, at the earliest possible time, transmit the message to the AMS HOPS.

6.3.8 Class 3 message processing

It is the POST application that manages the update of the ISAM. A sequence of Secure Data Frames sourced by either the AMS HOPS or the ISMS are delivered via the AMS HOPS to the POST and then applied to the ISAM. The ISAM delivers a number of Security Acknowledgements in return. Certain ISAM update processes are intended to be applied within a Controlled Environment see TS 1000-9 clause 11.2.5.

Class 3 messages received by a POST shall contain one or more ISAM Secure Data Frames destined to the POST's ISAM.

The POST shall transfer said Security Files to the ISAM, in the manner defined in ITSO TS 1000-7 and ITSO TS 1000-8.

For Uncontrolled Environments (see TS1000-9 clause 11.2.5.) where a number of update frames are to be applied in the execution of a single self-contained update then those frames shall be packed in order of the correct sequence in a single Class 3 ITSO message from the AMS HOPS.

6.3.8.1 Application of Secure Data Frames in an Uncontrolled Environment.

The provisions of this clause are mandated for all POSTs conforming to this version of the Specification.

6.3.8.1.1 Message Sequence Numbering

In order to ensure that the POST application only applies secure data frames from a class three message in the correct intra message sequence, the AMS function inserts two additional sequence numbers into class 3 messages using the HOPS to POST DTD (see TS1000-9).

Group updates and single ISAM updates belong to two separate series of sequential numbers that may be incremented without gaps or rollover. These sequential numbers may be read from the class 3 message and are determined by the AMS HOPS function in advance. The two sequence numbers shall be used in tandem by the POST application in determining whether to apply the Secure Data frames held in the message to the ISAM. Copies of the sequence number values of the most recently applied messages shall be held in NV storage on the ISAM.

The sequence numbers (which are unrelated to the File Transfer Sequence numbers found in the Secure Data frame itself) are defined in table 2 as follows:

Table 2 - Class 3 sequence numbers

XML tag	Contained in	Data Element Label	Data Type	# of bytes
ITSO_Group_Message_Seq_Number	Message	Group_Message_Seq#	HEX	3
ITSO_ISAM_Message_Seq_Number	Message	ISAM_Message_Seq#	HEX	3
N/A	ISAM	ISAM_Group_Seq#	HEX	3
N/A	ISAM	ISAM_Seq#	HEX	3

The ISAM_Group_Seq# in the ISAM is updated with the latest accepted copy of the Group_Message_Seq# and the ISAM_Seq# in the ISAM is updated with the latest accepted copy of the ISAM_ Message_Seq# as determined by the application logic defined in the following clause.

6.3.8.1.2 Normal Secure Data Frame Application logic and rules

The following logic shall be observed by the POST when applying Secure Data Frames from class 3 message

Note: Secure Data Frames that are solely for the purpose of updating the ISAM's Certified Date will be ignored by the logic in this clause and are managed in accordance with clause 6.3.8.1.3

Application Logic

Case 1 Update to single ISAM unconditional on Group updates

Where Group_Message_Seq# = 0

IF ISAM_Message_Seq# = 1+ ISAM_ Seq#: Execute INDEX_UPDATE function

FOR each Secure Data Frame in the message: Execute APPLY_FRAME function

NEXT

Flag the list of Secure Acknowledgements Logged against this message as a successful ISAM Update (Set the C_Flag to 0x01).

END IF

Case 2 Update to a Group unconditional on ISAM updates

Where ISAM_Message_Seq# = 0

IF Group_Message_Seq# = 1+ ISAM_Group_Seq#: Execute INDEX_UPDATE function

FOR each Secure Data Frame in the message: Execute APPLY_FRAME function

NEXT

Flag the list of Secure Acknowledgements Logged against this message as a successful ISAM Update

END IF

Case 3 Update depends on previously applied Group update

Where Group_Message_Seq# <= ISAM_Group_Seq#

IF ISAM_Message_Seq# = 1+ ISAM_ Seq# : Execute INDEX_UPDATE function

FOR each Secure Data Frame in the message: Execute APPLY_FRAME function

NEXT

Flag the list of Secure Acknowledgements Logged against this message as a successful ISAM Update.

END IF

Case 4 This case is no longer required.

Case 5 Personalising (Commissioning) an ISAM when installed in a POST

If both the Group Message Seq# and ISAM Message Seq# are set to zero in the message sent to the POST then the POST should ignore the Secure Data Frame checking rules and always apply the frame(s).

This case shall only be used on an ISAM already installed in a POST and only:

- During the ISAM personalisation (commissioning) phase (i.e from the initial ISMS_SREQ_ISAM_DATA to obtain the ISAMID, up to & including the ISMS_SREQ_KEYRING request, and any resulting Secure Data Frames).
- When updating an ISAM with the files needed to support Version 3 Class 3 messages (i.e if the POST supports LOG1 files/messaging, but the ISAM doesn't);
- If the ISAM's sequence numbers have become out of synchronisation with its managing AMS HOPS.

During execution of this case the POST application shall not use any of the files added.

Case 6 Retransmission of SECure ACKnowledgementS

Where Cases 1 to 5 inclusive do not apply, the contents of the candidate message shall not be applied to the ISAM.

If

The new message sequence number exceeds the current sequence number the POST shall respond to the first line HOPS with the latest SECACK_LIST held in the ISAM_LOG1 log file.

Else

The POST shall recover the SECACK_LIST relating to the candidate message and send it to the first line HOPS

The SECACK_LIST shall be packed in accordance with the final paragraph of clause 6.3.7

Note: The contents of the LOG_IDX file may be used by the POST to determine the Selection[note1] of the Secure Acknowledgement Log.

Functions

INDEX_UPDATE

Append the Start Of List (SOL) + End Of File (EOF) Marker (0xFFFF+ 0x0000) to the ISAM_LOG1 file. Store the values of the Group_Message_Seq# and ISAM_Message_Seq# as a new record in the LOG_IDX file having a Start_Index identical to the End_Index of the previous record and an End_Index pointing to the first byte of the new EOF. SET the completion flag C_FLAG to zero.

APPLY_FRAME:

Check for a match between the concatenated String ISAMID+FTS# (see TS1000-8 clauses 3.22.68 + 3.22.69) derived from the S(ISAMID) in the frame to be applied and any of the same strings found in a Selection[note 1] of the Secure Acknowledgement Log.

IF NO match is found: Apply the frame to the ISAM using the UPDATE_FRAME command.

IF an Acknowledgement is returned prefix it with SECACK_LEN then append this followed by the EOF Data Element to the ISAM_LOG1 file and update the End_Index in the LOG1_IDX file to point to the first byte of the EOF.

  IF the Acknowledgement is Unsuccessful[note 2]: Abort the Update

   END IF

END IF

END IF

ELSE do not apply the frame and prepare the matching secure ACK for retransmission to the HOPS in accordance with clause 6.3.7.

EXIT Function

Note 1: "Selection" is defined as that portion of the log file deemed relevant for searching.

Note 2: "Unsuccessful" is defined as

- the UPDATE_FRAME command returned an error

- the last script in the Security Acknowledgement did NOT return a success code (0x9000)

- the UPDATE_FRAME completion code did NOT return a success code (0x9000)

Note 3: The files required for logging and indexing are created by the AMS HOPS the structure content and access conditions are defined in Part 8 Annex F

Annex B herein gives an example of a typical progression of sequence numbering.

Application Rules

1. The POST shall ensure that if the application logic is satisfied all the Secure Data Frames in a single class 3 message shall be applied to the ISAM in the same order as they are packed in the message.

2. None of the Secure Data Frames from a new message shall be applied to an ISAM if one of the sequence numbers in the message is more than one greater than the equivalent number stored in the ISAM.

3. The POST shall not apply any further Secure Data Frames from a message once the application of a single Secure Data Frame has been unsuccessful.

4. The POST shall ensure that the last n Secure acknowledgements generated by the ISAM whether successful or not are stored in the ISAM.

5. The POST shall store in the ISAM a list of indices to the end of the last Secure Acknowledgement logged for the last m messages processed.

Note: n and m are determined by the AMS HOPS that manages the ISAM (or HSAM) in question.

6.3.8.1.3 DTS update Secure Data Frame Application logic and rules

Keep Alive frames shall have Group and ISAM sequence numbers set by the AMS HOPS that match the target ISAM. The TDF Data Element in the received frame shall have bit 2 set to a 1.

Note: Keep Alive frames shall not be mixed with other types of update frames in the same message.

Application logic

The following logic shall be observed by the POST when applying The Keep Alive Secure Data Frames from class 3 messages.

Case Keep Alive

Where there is only 1 Secure Data Frame in the message AND bit 2 of the TDF is set to 1, then:

- IF Group_Message_Seq# <= ISAM_Group_Seq# AND ISAM_Message_Seq# <= ISAM_ Seq#

  Apply the Keep Alive frame to the ISAM

- IF Group_Message_Seq# > ISAM_Group_Seq# AND/OR ISAM_Message_Seq# > ISAM_Seq#

  The POST shall respond to the first line HOPS with the latest SECACK_LIST held in the ISAM_LOG1 log file

  END IF

The SECACK_LIST shall be packed in accordance with the final paragraph of clause 6.3.7

Application Rules

1. The POST shall not modify the sequence numbers stored in the ISAM

2. The POST shall ensure that the Secure Data frames generated by the ISAM in acknowledgement "successful or otherwise" are returned as normal to the HOPS however it shall NOT store the ACK in the ISAM. Keep Alive frames are deemed benign to ISAM contents and multiple applications of the same or similar frames is permitted.

3. In the event that several Keep Alive updates are available only the most recent need be applied.

6.3.9 Sequence numbers

Every Data Frame contains a sequence number as described in ITSO TS 1000-6. However, sequence numbers are handled differently between Data Frames in a Class 1 message and Data Frames in Class 0 and Class 2 messages.

In a 'Class 1' Data Frame, the sequence number is generated by the ISAM as part of the Data Frame sealing process and this sequence number is linked to the IBatch Header.

In 'Class 0' and 'Class 2' Data Frames, the POST shall provide the sequence number. In the case of Data Frames that are used in Class 2 messages, it is important that this number is unique for all Data Frames that are pending ACKs as the ACK will contain that number as a reference. For Data Frames in Class 0 messages, sequence number uniqueness is not important.

6.4 Configuration handling

The POST shall be capable of receiving and processing the following configuration files from a HOPS:

- Hotlist
- Actionlist
- POST configuration data
- IPE Embodiment Parameters file   (if IPE creation is required)

ITSO does not mandate minimum storage requirements for these lists. Suppliers shall state the storage capacity (in records) of the POST for each of the above. These capacities shall be stated on the POST accreditation certificate.

POSTs shall only action data records contained in these configuration files if it can ascertain the validity of such records.

6.4.1 Hotlist

The ITSO Hotlist mechanism allows for the blocking of entities on the Media (ITSO Shell / IPE instances). ITSO Hotlists will be sent to a POST from its first-line HOPS.

The ITSO Environment and message set does not support the blocking of the customer Media itself. The highest level entity which may be blocked is the ITSO Shell.

6.4.1.1 Hotlist structure

The structure of the Hotlist is defined ITSO TS 1000-6. In summary, the list consists of one or more records, where each record is made up of a number of fields as follows:

- Header
- IPE identifier(s)   (optional)
- A Hotlist data group

6.4.1.2 Hotlist storage

Lists shall be processed as follows:

- The HOPS shall transmit each Hotlist record as a (signed) Class 2 Data Frame, which the POST shall check and process as defined in 6.3.6
- ITSO does not prescribe the storage format for the Hotlist, but does require that the format retains all data elements specified, and allows searching using the criteria specified in subsequent sub-clauses

6.4.1.3 Hotlist updates

For lists stored in the POST's own memory:

1. If the received HotListIdentifier equals the current HotListIdentifier AND RecordType = 1 or 5 append the received list to the current list;If the received HotListIdentifier equals the current HotListIdentifier AND RecordType = 1 or 5 append the received list to the current list;

2. If the received HotListIdentifier is greater than the current HotListIdentifier, taking into account rollover as defined in ITSO TS 1000-1 AND RecordType = 1 or 3: delete the current list and replace with the new list;

3. In all other cases do not delete or append to the current list but send an 0400 Exception message to the HOPS with ExceptionType set to a value of 148 as defined in ITSO TS 1000-6.

6.4.1.4 Hotlist searching

The POST shall support searching of the Hotlist by both:

• ITSO Shell reference	(key type 0) and
• IPE reference	(key type 1)

ITSO Shell reference (key type 0) searches shall be used whenever the presented Media platform carries a 'full' (version 1) ITSO Shell (see ITSO TS 1000-2 and ITSO TS 1000-10).

IPE reference (key type 1) searches shall only be used whenever the presented Media platform carries a 'compact' (version 2) ITSO Shell (see ITSO TS 1000-2 and ITSO TS 1000-10).

In both search mechanisms (key type 0, key type 1) the 'search string' generated by the POST shall consist of 8 bytes, which match the first 8 bytes of the header within the Hotlist record.

6.4.1.4.1 Key type 0

This search type shall be used when the prime reference is the ITSO Shell instance. The search string consists of the following elements, and is fully defined in ITSO TS 1000-6:

- Key type
- ITSO Shell iteration number
- IIN index
- ISRN (excluding IIN)

The full ISRN is not used. Instead of an explicit IIN value, an IIN 'reference index' is used. Both the HOPS and the POST shall contain a copy of the 'IIN Index Table', thus supporting this abbreviated mechanism. See 6.4.3.4 for further details.

6.4.1.4.2 Key type 1

This search type shall be used when the prime reference is an IPE instance. The search string consists of the following elements, and is fully defined in ITSO TS 1000-6:

- Key type
- IPE iteration number
- ISAM ID   (The ID of the ISAM that created the IPE instance)
- ISAM sequence number   (The sequence number for the IPE instance creation transaction)

6.4.1.5 Hotlist matching

If a match is found against a Hotlist search, then the POST shall:

- If HotType = 2, where an IPE ID Optional Additional Identification Group is included in the Hotlist record, then the POST shall use this information to identify the IPE that is the target of the Hotlist record
- Update the required data element(s) in the affected entity to indicate usage of this entity is blocked. The relevant entities are defined by the HotType and HotAction fields in the Hotlist record (see ITSO TS 1000-6)
- Use the services of the ISAM to seal these modified entities (i.e. ITSO Shell, Directory, IPE) and write these back to the Media
- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal those records and maintain the IBatch Header in an accurate state

- Signal to the POST operator the required action to be taken as regards to the Media / user. The action types are defined by the CustomerMediaDisposition field in the Hotlist record (see ITSO TS 1000-6)

6.4.1.5.1 ITSO Shell blocking

If the HotType and HotAction parameters require the ITSO Shell to be blocked, then the POST shall set the 'ShellBlocked' flag within the DirBitMap element of the Directory (see ITSO TS 1000-2).

6.4.1.5.2 IPE blocking

If the HotType and HotAction parameters require the IPE instance to be blocked, then the POST shall set the terminating SCT entry of that IPE according (see ITSO TS 1000-2).

6.4.1.5.3 Unblocking

An entity that has been blocked may be 'un-blocked' by either of the following processes:

- Via an Actionlist action (for Shells only) or
- Via a manual process at the POST

  ITSO does not mandate that POSTs must support the manual process method. During the unblocking process (whether manual or in response to an Actionlist item) the POST shall carry out the following actions:

- Increment the relevant iteration number (ITSO Shell or IPE) within the affected entity. In instances where the Customer Media supports software anti-tear, all copies of the Directory or Value Records shall be incremented. The new iteration number shall be used within the Transaction Records generated (see below)

- Use the services of the ISAM to seal the modified entity and write this back to the Media

- Verify that all data written to the Media was carried out correctly

- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

POSTs shall not unblock, and shall not increment the Shell or IPE iteration number (as appropriate), of any Shell or IPE that is not currently blocked (irrespective of whether the unblock function is instigated by a manual instruction or an Actionlist item).

6.4.1.6 Hotlist Messaging Errors

In circumstances where a POST cannot process a received Hotlist, the POST shall respond with a Negative Acknowledge to a Class 2 Message (NAK2), using either reason code 0x09 or 0x0A as appropriate.

6.4.2 Actionlist

The ITSO Actionlist mechanism allows for a central facility (e.g. a HOPS) to define actions that will be remotely executed on a given instance of Customer Media when such Media is presented to a POST. In concept, this is a wider scoped variation on the Hotlist functionality defined previously. Like Hotlists, Actionlists will be sent to a POST from its first-line HOPS.

6.4.2.1 Actionlist structure

The structure of the Actionlist is defined ITSO TS 1000-6. In summary, the list consists of one or more records, where each record is made up of a number of fields as follows:

- Header
- IPE identifier   (optional)
- An Actionlist data group
- Actionlist data elements   (depending on the type of action)

6.4.2.2 Actionlist storage

Lists shall be processed as follows:

- The HOPS shall transmit each Actionlist record as a (signed) Class 2 Data Frame, which the POST shall check and process as defined in 6.3.6
- ITSO does not prescribe the storage format for the Actionlist, but does require that the format retains all data elements specified, and allows searching using the criteria specified in subsequent sub-clauses

6.4.2.3 Actionlist updates

For lists stored in the POST's own memory:

1. If the received ActionListIdentifier equals the current ActionListIdentifier AND RecordType = 2 or 6 append the received list to the current list;

2. If the received ActionListIdentifier is greater than the current ActionListIdentifier, taking into account rollover as defined in ITSO TS 1000-1, AND RecordType = 2 or 4: delete the current list and replace with the new list;

3. In all other cases do not delete or append to the current list to the HOPS with ExceptionType set to a value of148 as defined in ITSO TS 1000-6.

6.4.2.4 Actionlist searching

The POST shall support searching of the Actionlist by both:

• ITSO Shell reference	(key type 0) and
• IPE reference	(key type 1)

ITSO Shell reference (key type 0) searches shall be used whenever the presented Media platform carries a 'full' (version 1) ITSO Shell (see ITSO TS 1000-2 and ITSO TS 1000-10).

IPE reference (key type 1) searches shall only be used whenever the presented Media platform carries a 'compact' (version 2) ITSO Shell (see ITSO TS 1000-2 and ITSO TS 1000-10).

In both search mechanisms (key type 0, key type 1) the 'search string' generated by the POST shall consist of 8 bytes, which match the first 8 bytes of the header within the Actionlist record.

When checking an Actionlist at the time a CM is presented, it is recommended that a POST checks for all action items relating to the presented Shell, and does not stop searching the list after the first item is found. Licensed Members shall note that where a POST is configured to stop searching the list after the first relevant item is found, but there are multiple actions for the current Shell in the list, then the action processed may not be that required for the current journey, resulting in a poor customer experience. POSTs shall provide the capability to either stop after the first action item is found in the list, or to continue to search the

list until all relevant action items have been found, as an operator configurable option. Where multiple action items are found, all the action items relevant to the presented Shell shall be processed during a single Transaction, irrespective of whether an individual action can or cannot be applied.

6.4.2.4.1 Key type 0

This search type shall be used when the prime reference is the ITSO Shell instance. The search string consists of the following elements, and is fully defined in ITSO TS 1000-6:

- Key type
- ITSO Shell iteration number
- IIN index
- ISRN (excluding IIN)

The full ISRN is not used. Instead of an explicit IIN value, an IIN 'reference index' is used. Both the HOPS and the POST shall contain a copy of the 'IIN Index Table', thus supporting this abbreviated mechanism. See 6.4.3.4 for further details.

6.4.2.4.2 Key type 1

This search type shall be used when the prime reference is an IPE instance. The search string consists of the following elements, and is fully defined in ITSO TS 1000-6:

- Key type
- IPE iteration number
- ISAM ID   (The ID of the ISAM that created the IPE instance)
- ISAM sequence number   (The sequence number for the IPE instance creation transaction)

6.4.2.5 Actionlist matching

If a match is found against an Actionlist search, then the POST shall:

- If the matching Actionlist record relates to an IPE, where an IPE ID Optional Additional Identification Group is included in the Actionlist record, then the POST shall use this information to identify the IPE that is the target of the Actionlist record
- Update the required data element(s) in the affected entity as required by the action codes in the relevant Actionlist record. See ITSO TS 1000-6 for full details of the available actions
- Update the ActionSequenceNumber within the IPE (see below)
- Use the services of the ISAM to seal these modified entities (i.e. ITSO Shell, Directory, IPE) and write these back to the Media
- Verify that all data written to the Media was carried out correctly
- Generate all required Transaction Records (including exception records if required) and use the services of the ISAM (as defined in ITSO TS 1000-7 and ITSO TS 1000-8) to seal said records and maintain the IBatch Header in an accurate state

6.4.2.5.1 Action sequence number

The ActionSequenceNumber field stored in the IPE instance on the central IPA prevents the application of the same action to an IPE instance more than once. This field also enforces the correct sequencing of action events.

When the POST attempts to action an Actionlist match, if the ActionSequenceNumber within the record is not equal to the ActionSequenceNumber in the IPE instance, then the action shall not be carried out.

If the ActionSequenceNumber within the record is equal to the ActionSequenceNumber in the IPE instance, then the action shall be carried out and the ActionSequenceNumber in the IPE instance shall be incremented. This new ActionSequenceNumber shall be returned in the Transaction Records for the action.

Note: ITSO Shells do not contain an ActionSequenceNumber.

Note: Prior to version 2.1.4. of the specification ITSO did not provide a mechanism which prevents Actionlist items which create IPEs (where ActionToTake = 1) from being actioned more than once. Prevention of multiple actioning was the responsibility of the Product Owner. Clause 6.4.2.6 was added to provide for an alternative methodology which overcomes this limitation.

6.4.2.6 IPE_Fulfilment_Action Matching

If a match is found against an Actionlist search that includes an IPE_Fulfilment_Action (i.e. ActionToTake = 15 dec) then a POST shall apply the following rules:

1. The POST shall apply the actions described in TS1000-6 clause 5.3.3 in same order as the tagged objects present within the IPE_Fulfilment_Action.

2. Wherever the action described for a tagged object cannot be executed successfully the entire IPE_Fulfilment_Action shall be aborted and the CM left in the state it was before the IPE_Fulfilment_Action was attempted.

3. A complete update sufficient to satisfy the customer's requirements and consistent with the commercial arrangements shall be contained in a single IPE_Fulfilment_Action.

4. Where the total number of deletions and additions together does not exceed 4 the IPE_Fulfilment_Action may be completed successfully within a single Transaction.

5. Where the total number of deletions and additions together exceeds 4 and any IPEs deleted are not critical to the future operation of the CM then the IPE_Fulfilment_Action shall be split into more than 1 CM session as follows:

    A. Firstly up to 4 deletions per CM session for as many sessions as required.

    B. Secondly up to a maximum of 4 additions in a single session only.

6. After completion of the IPE_Fulfilment_Action the POST application shall carry out any other action list items present for IPEs that have just been added to this Shell.

7. Transaction records for every IPE deleted and added by the IPE_Fulfilment_Action and where modified subsequently shall be produced as normal.

6.4.2.7 Actionlist Messaging Errors

In circumstances where a POST cannot process a received Actionlist, the POST shall respond with a Negative Acknowledge to a Class 2 Message (NAK2), using either reason code 0x09 or 0x0A as appropriate.

6.4.3 POST configuration data

The ITSO POST configuration data mechanism allows for a central facility (e.g. a HOPS) to define the parameters that are required for Interoperability. Like Hot and Actionlists, POST configuration data parameters will be sent to a POST from its first-line HOPS.

The following rules shall apply for the receiving of PCD by a POST

1. The POST application shall request the delivery of the PCD for that POST in all or part from a HOPS at any time by sending an 0803 message

2. The POST application shall, when able, receive all or part of its PCD from a HOPS or supervisory POST in the event that an update is sent

Where the downstream delivery channel involves partial delivery of a complete business update the POST application shall not action the update until a Manifest is received and has been verified.

The following application rules shall apply to the POST

1. The POST shall verify that it has a copy of all the PCD intended for it as follows:

   - By verifying the 0Bxx_TableHash in the manifest agrees with the Hash of the contents of the Parameter Table 0Bxx as calculated by the POST

   - By further verifying that the ManifestHash agrees with the Hash of the contents of the Manifest as calculated by the POST

   - Ensure that the Manifest_DTS postdates the current manifest

2. In the event that the POST cannot verify the contents of its PCD are complete and correct then that POST shall as determined by the scheme rules:

   - Primarily default to using an earlier complete and correct version of the PCD

   - Under exceptional circumstances cease to accept ITSO CM

   - After successful verification of both of the above cases the POST shall ACK the message otherwise a NAK is sent

3. Where the PCD does include index numbering the POST shall search table rows in ascending order of IndexNumber and the note below shall not apply.

Note: Where the PCD does not include index numbering the POST shall search table rows in row order first to last.

6.4.3.1 POST configuration data structure

The structure of the POST configuration data is defined ITSO TS 1000-6.

6.4.3.2 POST configuration data storage

The POST shall store the received POST configuration data within its own secure non-volatile memory. ITSO does not prescribe the storage format for the list.

6.4.3.3 POST configuration data updates

If the ParameterTableIdentifier in a received POST configuration data record is greater than that of the identifier of the associated parameter table currently stored (taking roll-over into account), then the POST shall clear down that parameter table. It shall then start a new parameter table, the first row of which shall be the newly received record.

If the ParameterTableIdentifier in a received POST configuration data record is the same as the identifier of the associated parameter table currently stored (taking roll-over into account), then the POST shall append the POST configuration data record as a new row in that parameter table.

6.4.3.4 POST configuration data parameter tables

POSTs shall support and use the POST configuration data parameter tables defined in ITSO 1000-6.

6.4.3.5 Handling multiple row matches

There will be circumstances where the match criteria of two or more rows are the identical, and where there is overlap in the effective date range defined by StartDateTime and EndDateTime. In these circumstances the POST shall apply the parameters contained in the first row found in the table.

Note that creators of parameter tables are advised to avoid the situation where there is overlap in the effective date range defined by StartDateTime and EndDateTime.

6.4.3.6 Application of POST Configuration Data

When a POST finds an acceptable IPE Instance in a presented ITSO Shell, then it shall check to see whether any POST configuration data applies to that IPE's embodiment. If it finds such POST configuration data, then it shall apply that data when processing that IPE Instance.

6.4.4 IPE Embodiment Parameters

The IPE Embodiment mechanism allows for a central facility (e.g. a HOPS) to define the parameters that are required for IPE creation.

6.4.4.1 IPE Embodiment Parameters file structure

The structure of the IPE Embodiment Parameter file is defined ITSO TS 1000-6. In summary, the file consists of one or more records, where each record is made up of a number of fields as follows:

- Element number

- Rule code

- Data size

- Data

The HOPS sends each record as a (signed) Class 2 Data Frame, which the POST shall check and process as defined in 6.3.

6.4.4.2 IPE Embodiment Parameters file storage

The POST shall store the received embodiment parameter file within its own secure non-volatile memory. ITSO does not prescribe the storage format for the list.

6.4.5 Miscellaneous POST to HOPS and HOPS to POST Messages.

The POST shall support the reception and transmission of these messages as indicated for each message code.

6.4.5.1 Physical ISAM Installation Notification message Code 0803

The link and functional relationship between ISAM, POST and location is established by the Service operator then reported to the AMS HOPS. This linking and location process is mainly driven by manual processes however this message shall be used to inform the AMS HOPS when a particular ISAM has been installed in a POST device. This will allow the accurate updating of the POST to ISAM relationship as maintained by the AMS HOPS.

As a minimum requirement this message shall be sent as a signed class 2 Data Frame by the POST to the AMS HOPS when:

- one of the parameters embodied in the message has changed; or

- more frequently or at other times as scheme management rules dictate.

Note that POST configuration must be set up correctly in respect of data included in the 0803 message. HOPS are very likely to change their internal configuration in response to the data contained in the message, and if that data is not correct then problems are likely to arise.

The structure and content of this message is defined in TS1000-6.

6.4.5.2 ISAM inventory message codes 080A and 080B

These messages allow one or more ISAM Parameter Table files to be read from the ISAM and returned to the (first line) HOPS. The HOPS requests one or more files using message code 080A. The POST shall return the requested files together with a status code for each file requested using message code 080B.

The POST shall check the originator is its first line HOPS, by checking the originating ISAM ID. Should the originating ISAM ID not match that of the first line HOPS, then the POST shall reject the message and respond to the HOPS with a NAK.

The POST shall read, using ISO commands Select and Read, the requested ISAM files and return them to the First Line HOPS as defined in ITSO TS1000-6. Only freely readable ISAM files shall be returned to the HOPS. Any errors received when attempting to read a file due to non-existence, permission errors etc will be reported using the Status word from the ISAM in field DF62 of the 080B response message. After reading, care should be taken to reselect the ITSO MF.

The minimum granularity for the response shall be at file level and shall include individual transaction files. Hash sealed response messages shall be used. Both request and response shall be encoded using ASN.1 TLV format, using data elements defined in ITSO TS1000-6.

In circumstances where the POST is unable to respond with an 080B message, then the POST shall respond with a Negative Acknowledgement to a Class 2 Message (NAK2), using reason code 0x08 indicating that the POST was unable to process an 080A message.

Annex A normative Customer Information Messages
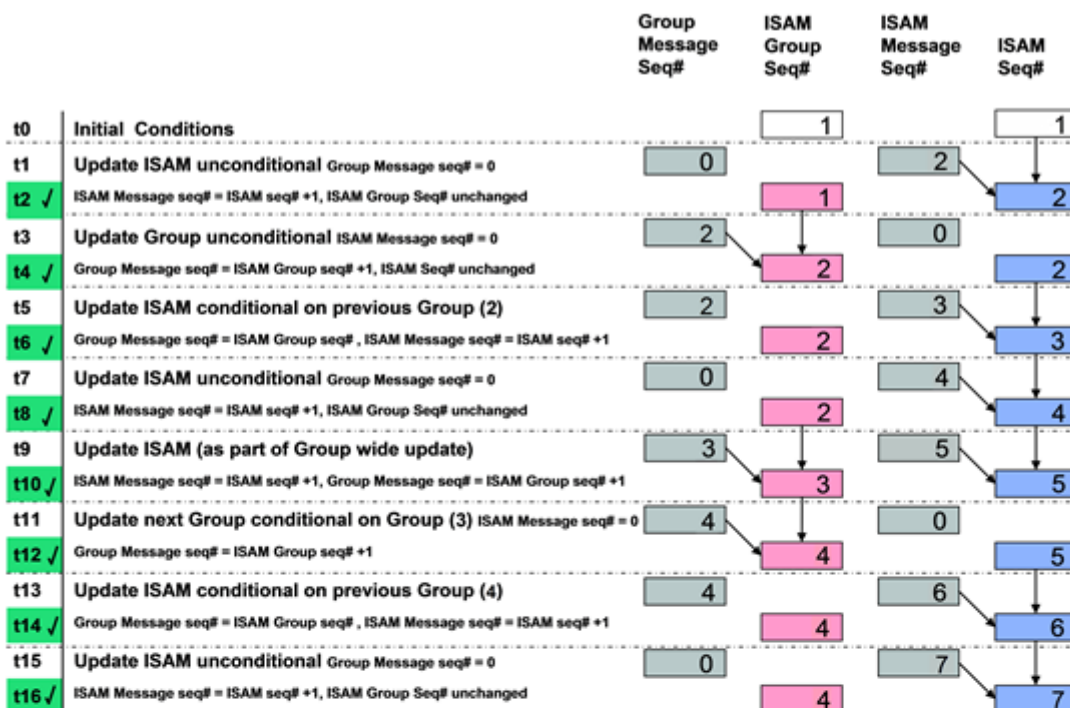
Table A.1 – Customer messages

Table A.1 - Customer messages

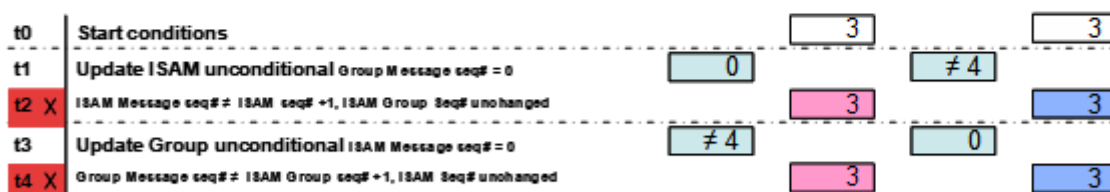| Condition	ITSO Customer Message
POST In Service	Ready
POST Out of service	Out of service
ITSO Shell CRC failed	Seek assistance
ITSO Shell expired	Seek assistance
ITSO Shell blocked	Seek assistance
FVC not supported / authorized	Seek assistance
KSC not supported / authorized	Seek assistance
Invalid Seal on both Directories	Seek assistance
Product candidate list is empty	Seek assistance
No authorized products found	Seek assistance
Product selection by Customer Media holder required	Please select product to use
Selected product not valid @ time or place	Seek assistance
Selected product cannot cover cost of journey	Seek assistance
Successful transaction	OK
Action applied	Update successful
Auto-Renew applied	Update successful
Media prematurely removed	Please replace Media

Annex B Informative Typical Physical ISAM Group and ISAM sequence number updating
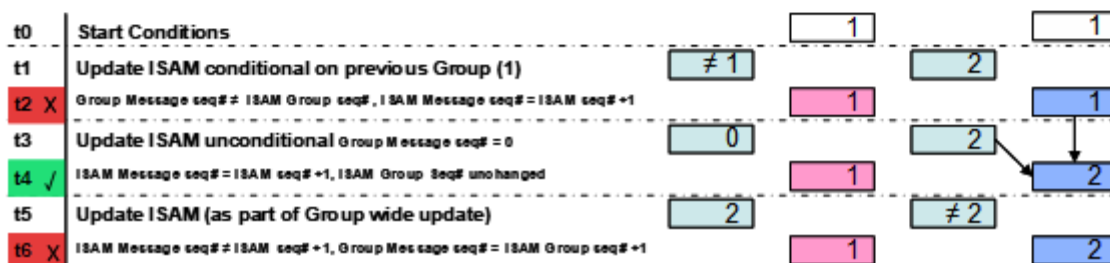
Figure B.1 Example sequence number progression

Figure B.1 - Example sequence number progression