

<b>Issuing Authority:</b>	<b>Owner:</b>	<b>Project Editor:</b>
ITSO	Technology at ITSO	Mike Eastham
<b>Document number</b>	<b>Part Number:</b>	<b>Sub-Part Number</b>
ITSO TS 1000	2	
<b>Issue number (stage):</b>	<b>Month:</b>	<b>Year</b>
2.1.3	April	2008
<b>Title:</b>		
ITSO TS1000-2 <i>Interoperable public transport ticketing using contactless smart customer media – Part 2: Customer media data structure</i>		
<b>Replaces Documents:</b>		
ITSO TS 1000-2 2007-06 issue number 2.1.2		

## Revision history of current edition

Date	ITSO Change Ref.	Editor ID	Nature of Change to this Document (or Part)
Dec 2002	DCI 100 / create v2.1	SLB	Incorporate TN0001 and reformat. Remove Annex A. Renumber and fix cross-references.
Feb 2003	Ditto	CJS/SLB	Complete re-write to align with IOPTA proposals and current status of EN 1545. Card and crypto format definitions moved to Part 10. Formatted as committee draft.
April 2003	Ditto	CJS/SLB	Minor corrections implemented. Clause 9 moved to Part 10. Formatted as final draft.
Oct 2003		CJS / SLB	Removal of Transient Ticket Data Group and replace with Log Directory Entries. Added Orphan IPE Data Group and Cyclic Log. Removed software Anti-tear. Incorporated accepted review comments including change from card to Customer Media. Format and prepare for issue at 2 <sup>nd</sup> CD.
Nov 2003		CJS / SLB	Implementation of Author's change instruction (TC N0335) to resolve comments and speed up directory access for certain CMDs.
Nov 2003		CJS / SLB	Implement Author's 2 <sup>nd</sup> change instruction (TC N0337). Fix editorial comments on 3 <sup>rd</sup> CD. Issue 4 <sup>th</sup> CD.
Nov 2003		SLB	Editorial changes only. Issue 1 <sup>st</sup> consultation draft.
Jan 2004		JH	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 include ISADs following approval by DfT
Feb 2008		CJS	Updated to include ISADs following approval by DfT
Apr 2008		MPJE	Final Editing before Publication

Document Reference: **ITSO TS 1000-2**

Date: 2008-04-21

Version: 2.1.3

Ownership: ITSO

Secretariat: Technology at ITSO

Project Editor: Mike Eastham

## **ITSO Technical Specification 1000 – Interoperable public transport ticketing using contactless smart customer media – Part 2: Customer Media data and Customer Media architecture**

ISBN: 978-0-9548042-3-7

"Published for the Department for Transport under licence from the Controller of Her Majesty's Stationery Office. The Department for Transport, its officials, Ministers and the Secretary of State for Transport do not guarantee the accuracy, completeness or usefulness of this information; and cannot accept liability for any loss or damages of any kind resulting from reliance on the information or guidance this document contains.

© Queen's Printer and Controller of Her Majesty's Stationery Office, 2008.

Copyright in the typographical arrangement and design rests with the Queen's Printer and Controller of Her Majesty's Stationery Office.

For any other use of this material please apply for a Click-Use Licence at [www.opsi.gov.uk/click-use/index.htm](http://www.opsi.gov.uk/click-use/index.htm), or by writing to the Licensing Enquiries, Information Policy Division, Office of Public Sector Information, St Clements House, 2-16 Colegate, Norwich NR3 1BQ, fax 01603 723000, or e-mail [HMSOlicensing@cabinet-office.x.gsi.gov.uk](mailto:HMSOlicensing@cabinet-office.x.gsi.gov.uk).

This publication, excluding logos, may be reproduced free of charge in any format or medium for research, private study or for circulation within an organisation. This is subject to it being reproduced accurately and not used in a misleading context. The material must be acknowledged as copyright of the Queen's Printer and Controller of Her Majesty's Stationery Office, and the title of the publication specified."

## 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](http://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 smart 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 ..... 7**

**1.1 Scope of Part 2..... 7**

**2. The ITSO Shell architecture..... 8**

**2.1 Data Groups and Cyclic Log ..... 8**

**2.2 Data Structures ..... 9**

**2.3 Data Elements ..... 9**

**2.3.1 The Dataset structure..... 10**

**2.4 ITSO Shell overview ..... 10**

**2.4.1 Manufacturer's ID (MID) ..... 10**

**2.4.2 ITSO Shell Environment Data Group ..... 11**

**2.4.3 Directory Data Group ..... 11**

**2.4.4 IPE Data Groups ..... 11**

**2.4.5 Orphan IPE Data Groups ..... 12**

**2.4.6 Value Record Data Groups ..... 12**

**2.4.7 General requirements ..... 12**

**2.4.8 The Cyclic Log ..... 13**

**3. The Customer Media architecture..... 15**

**3.1 CM electrical and physical characteristics ..... 15**

**3.2 CM memory organisation..... 15**

**3.3 Anti-tear protection ..... 16**

**3.3.1 Mechanisation of Anti-tear protection..... 16**

**3.4 Relationships between CM, ITSO Shell, applications and IPEs ..... 18**

**3.5 Access to the Shell..... 19**

**3.5.1 Selecting the ITSO Shell ..... 19**

**3.5.2 Determining the CM..... 20**

**3.5.3 Accessing the CM layer number ..... 20**

**3.5.4 Security..... 21**

**4. The ITSO Shell Environment Data Group..... 22**

**4.1 ITSO Shell Environment Data Group Dataset..... 22**

**4.1.1 ShellLength .....23**

**4.1.2 ShellBitMap .....23**

**4.1.3 ShellFormatRevision .....23**

**4.1.4 ITSO Shell Reference Number (ISRN).....23**

**4.1.5 Format Version Code (FVC).....24**

**4.1.6 Key-Strategy Code (KSC).....24**

**4.1.7 Key-set Version Code (KVC).....24**

**4.1.8 ITSO Shell EXPIry date (EXP) .....24**

**4.1.9 Size of memory Sector (B) .....24**

**4.1.10 Number of Sectors (S).....24**

**4.1.11 Maximum number of Directory Entries (e#).....25**

**4.1.12 Size in bytes of the Sector Chain Table (SCTL) .....25**

**4.1.13 Optional: Multi-application CM reference (MCRN).....25**

**4.1.14 Padding.....25**

**4.1.15 ITSO Shell Environment Checksum (SECRC) .....25**

**4.2 Compact ITSO Shell Environment Dataset .....25**

**5 The Directory Data Group .....26**

**5.1 Directory Data Group Dataset .....26**

**5.1.1 DIRLength.....26**

**5.1.2 DIRBitMap.....27**

**5.1.3 DIRFormatRevision .....27**

**5.1.4 Directory Entries .....27**

**5.1.5 Sector Chain Table array (SCT).....28**

**5.1.5.5 Sector Chain Table Data Elements (SCT(i)) relating to the cyclic log.....31**

**5.1.6 Directory Sequence Number (DIRS#) .....34**

**5.2 Directory Instance Identifier (InstanceID) .....35**

**5.2.1 Key Identifier (KID) .....35**

**5.2.2 ITSO Shell Iteration number (INS#).....35**

**5.2.3 ITSO Security Application Module Identity (ISAMID) .....35**

**5.3 Directory Data Group Seal .....37**

**6. The IPE Data Group .....38**

**6.1 IPE Directory Entry structure .....38**

**6.1.1 Coding of an IPE Directory Entry ..... 38**

**6.1.2 OID Extension Flag (EF)..... 39**

**6.1.3 Operator Identification (OID) ..... 39**

**6.1.4 IPE Type (TYP) ..... 39**

**6.1.5 IPE Sub Type (PTYP) ..... 39**

**6.1.6 Value Group Present flag (VGP)..... 39**

**6.1.7 IINL ..... 40**

**6.1.8 Expiry (EXP) ..... 40**

**6.2 IPE Dataset ..... 40**

**6.2.1 IPELength ..... 40**

**6.2.2 IPEBitMap ..... 40**

**6.2.3 IPEFormatRevision ..... 40**

**6.2.4.IPE Data Elements ..... 40**

**6.2.5 Padding..... 41**

**6.2.6 Optional IIN..... 41**

**6.3 IPE InstanceID..... 41**

**6.3.1 Key Identifier (KID) ..... 41**

**6.3.2 IPE Iteration number (INP#) ..... 41**

**6.3.3 ISAM Identity (ISAMID) ..... 42**

**6.3.4 ISAM Sequence number (ISAMS#)..... 42**

**6.4 IPE Data Group Seal ..... 42**

**6.5 Orphan IPE Data Groups ..... 42**

**6.5.1 The Orphan IPE Data Group as an IPE ..... 43**

**6.5.2 The Orphan IPE Data Group as an envelope for a Transient Ticket Record ..... 43**

**7 Value Record Data Group ..... 45**

**7.1 Value Record Directory Entry ..... 45**

**7.2 Value Record Dataset structure ..... 45**

**7.2.1 VGLength..... 46**

**7.2.2 VGBitMap..... 46**

**7.2.3 VGFormatRevision ..... 47**

**7.2.4. Value Record Data Elements (VR(i)) ..... 47**

**7.3 Value Record InstanceID structure..... 48**

**7.3.1 Coding of the Value Record InstanceID .....48**

**7.4 Value Record Seal .....48**

**8 Log Directory Entries .....50**

**8.1 The Log Directory Entry .....50**

**8.1.1 Coding of the Log Directory Entry .....50**

**8.1.2 Log pointer flag (LPF) .....50**

**8.1.3 Pointer (PTR).....50**

**8.1.4 Entry / Exit Indicator (EEI).....51**

**8.1.5 Date Time Stamp (DTS) .....51**

**8.1.6 Passback Time (PTLBM) .....51**

**Annex A (informative) CRC Generation .....52**

**A.1 Example of CRC generation.....52**

**A.1.1 Examples of bit patterns .....52**

**A.1.2 Code sample written in C language for CRC calculation .....53**

**Annex B (normative) OID numbering .....55**

**B.1 Roles of organisations .....55**

**B 1.1 Initial OID range.....55**

**B 1.2 New ranges supported .....56**

**B 1.3 Mechanisation of the extended numbering system .....56**

## 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 components (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 2

This Part of ITSO TS 1000 defines the ITSO Shell and data storage. In particular it defines:

- the ITSO Shell architecture;
- the Customer Media architecture;
- the ITSO Shell Environment Data Group;
- the Directory Data Group;
- the IPE Data Group;
- the Value Record Data Group;
- Log Directory Entries.

This Specification uses, as its base, the Data Elements and Data Structures defined in the emerging European Standards EN1545 and IOPTA referenced in the bibliography in ITSO TS 1000-1.

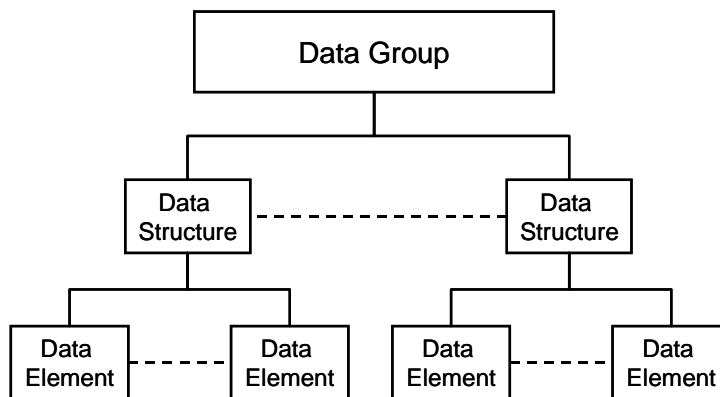
Throughout this Part of ITSO TS 1000 any references to the Customer Media Definition (CMD) refer to information contained within ITSO TS 1000-10 which maps all ITSO Data Groups to a particular Customer Media (CM) platform.

## 2. The ITSO Shell architecture

The ITSO Shell is a notional area that holds the collection of all ITSO related data. The ITSO Shell may occupy a given Customer Media (CM) platform exclusively or co-exist with other functions.

All data held in the ITSO Shell is constructed from Data Elements that are concatenated to form Data Structures, which in turn are part of a Data Group.

The hierarchy of Data Groups, Data Structures and Data Elements that together make up the ITSO Shell is illustrated in Figure 1.



**Figure 1 – Hierarchy of data within the ITSO Shell**

### 2.1 Data Groups and Cyclic Log

The following Data Group types and Cyclic Log make up the ITSO Shell:

- |   |                                   |                                                                                                                                                                                                                                                                                                                                                                  |
|---|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | ITSO Shell Environment Data Group | One copy per ITSO Shell. Holds ITSO Shell Owner, unique ITSO Shell sequence number, optional multi-application card issuer number and other ITSO Shell structure details. It remains fixed for the life of the ITSO Shell.                                                                                                                                       |
| 2 | Directory Data Group              | One or two copies <sup>1</sup> per ITSO Shell. It holds entries for every ITSO Product Entity (IPE) and Log held in the ITSO Shell and points to their location. It may change frequently throughout the life of the ITSO Shell.                                                                                                                                 |
| 3 | IPE Data Group                    | As many as are loaded into the ITSO Shell at any one time. IPE Data Groups are unique instances of IPEs. IPE Data Group contents may change periodically throughout the life of the IPE. IPE Data Groups may be added and removed throughout the life of the ITSO Shell. IPE Data Groups will usually make up the majority of Data Groups within the ITSO Shell. |
| 4 | Value Record Data Group           | Where frequently changing values are required these are stored in one or two copies <sup>1</sup> of a Value Record Data Group linked to the required IPE Data Group. Value Record Data Group contents may change frequently throughout the life of the associated IPE.                                                                                           |
| 5 | Cyclic Log                        | One copy per ITSO Shell if required. The Cyclic Log holds records of temporary information such as tickets or events.                                                                                                                                                                                                                                            |

---

<sup>1</sup> Depending on the Anti-tear method employed

## 2.2 Data Structures

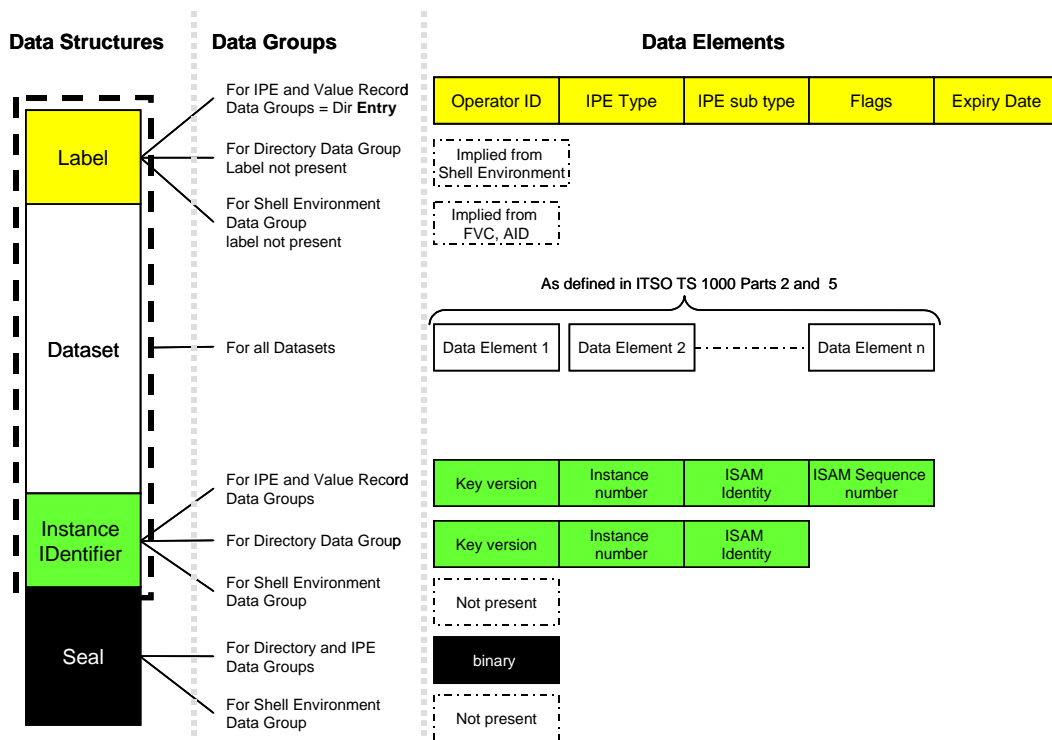
Note: Throughout this Part of ITSO TS 1000 commonly used Data Structures are colour coded as an aid to recognition.

Data Groups consist of up to four Data Structures. The Data Structures fall into categories as follows:

- 1      The Label                                      The title of the Data Group, for IPE and Value Record Data Groups. In this case the Label is also normally the Directory Entry.  
— In the case of the ITSO Shell Environment and Directory Data Group the Label is not present but implied.
  
- 2      The Dataset                                    A concatenation of Data Elements that make up the main content of all the Data Groups.
  
- 3      The Instance IDentifier                      A concatenation of Data Elements that identify the creator of the Data Group. The contents of the Instance IDentifier are context sensitive and in the case of IPE and Value Record Data Groups also uniquely identify a particular instance of the Group.  
— The Instance Identifier is not present in the Shell Environment Data Group.
  
- 4      The Seal                                         A context sensitive Data Element cryptographically derived from the contents of the other structures within the associated Data Group.  
— The Seal is not present in the ITSO Shell Environment Data Group.

## 2.3 Data Elements

Data Elements defined throughout ITSO 1000 when concatenated together make up the Data Structures. Figure 2 illustrates the common Data Elements and Data Structures within the various Data Groups.



**Figure 2 - Common Data Structures, Groups and Elements**

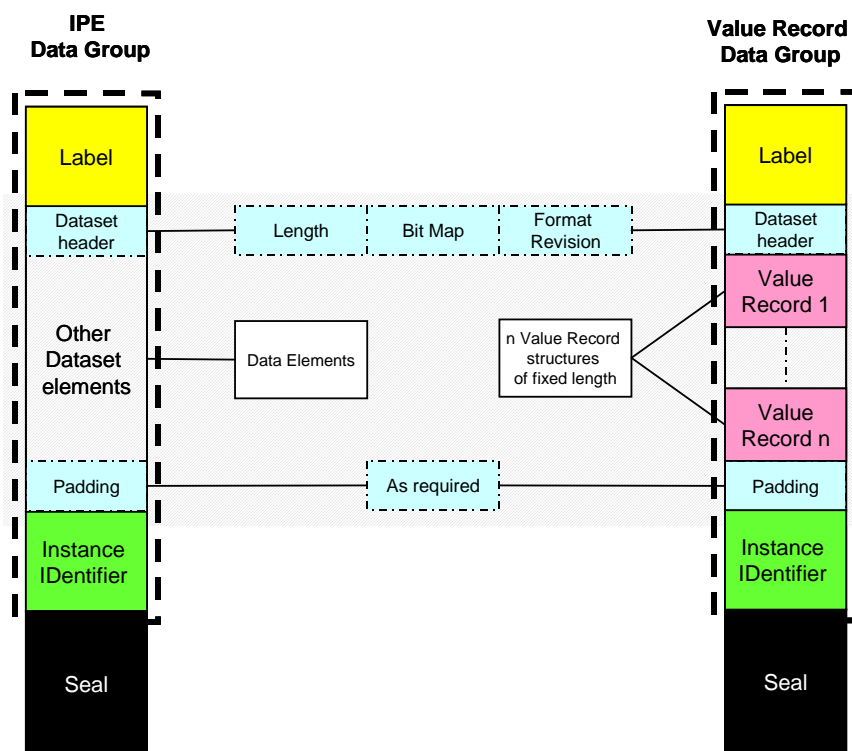
Detailed definitions of these Data Structures and Data Groups are to be found in later clauses of this Specification.

### 2.3.1 The Dataset structure

Datasets are concatenations of Data Elements. The first three and the last Data Element in the Dataset have special significance. The first three Data Elements are collectively known as the Dataset Header and are common to all Data Groups. The Dataset Header Data Elements contain:

- the length of the Dataset in Blocks;
- a Bit-Map defining which and how many optional Data Structures shall be present;
- the format revision number of the Dataset.

The final Data Element of the Dataset is included where needed for padding the Dataset to a whole number of bytes consistent with the Block size specified. Figure 3 illustrates this construction for an IPE Data Group and a Value Record Data Group. See clauses 6 and 7 respectively for full details of the make up of these Data Groups.



**Figure 3 - Dataset construction within the Data Groups**

A Value Record Data Group shall contain at least 2 Value Records.

All Value Records within a given Value Record Data Group shall be of the same length. See ITSO TS 1000-5.

### 2.4 ITSO Shell overview

The ITSO Shell architecture facilitates time efficient implementation of transport applications on various types of CM ranging from memory only media to microprocessor based media.

This sub-clause covers the Data Structures and a key Data Element present in a typical ITSO Shell.

#### 2.4.1 Manufacturer's ID (MID)

Many CM types carry a unique serial number encoded into them during manufacture and fixed for the life of the CM. Where the MID is readily accessible it shall be used in conjunction with the ITSO Shell Reference Number (ISRN) in binding the ITSO Shell to the CM. In the event that the MID is not readily accessible then the ISRN shall be used in lieu of the MID providing the CM has been authenticated.

Refer to ITSO TS 1000-10 for details on how to ascertain the MID.

### 2.4.2 ITSO Shell Environment Data Group

Within the ITSO Shell Environment Data Group are version codes that shall be used by the POST to decode:

- the formatting of Data Groups on different CM;
- the security algorithms used by the CM;
- the key sets used by the CM.

The ITSO Shell Environment Data Group also contains information about the structure of the CM and Directory Data Group.

Where required the number assigned by the issuer of a multi-application card that hosts an ITSO Shell may optionally be copied within the ITSO Shell.

ITSO Shell Environment Data Group content shall remain fixed for the life of the ITSO Shell.

### 2.4.3 Directory Data Group

The Directory Data Group contains sufficient information for basic IPE acceptance and points to IPE Data Groups, which may then be opened, if required, as part of the transaction. The Dataset of the Directory Data Group is termed the Directory.

Entries in the Directory include details of the owner and type of IPE present, along with an IPE validity date, and point via the Sector Chain Table (SCT) to the location where the IPE Data Group resides. Interpretation of the Sector Chain Table (see sub-clause 5.1) indicates whether the IPE has been: first used; is blocked; and which copy of any associated Value Record Data Group is current.

Special Directory Entry formats shall be used to indicate the presence of:

- Private (non ITSO) Applications or Products;
- a Cyclic Log.

The option to include within the Directory one or more entries for Private Applications or Products allows memory space to be allocated for non-ITSO use within the ITSO Shell. This option offers existing transport systems a migration path to ITSO and the opportunity for new, non-ITSO, Products to capitalise on the use of the ITSO architecture.

There are two types of Log Entry that may be present in the Directory that facilitate the mechanisation of simple closed system entry and Anti-passback logging and / or management of Transient Ticket Records and closed system entry records using a Cyclic Log.

The Directory shall incorporate Anti-tear protection allowing the most recent update to be identified and verified whilst being able to recover the previous state in the event of a torn transaction.

The Seal of the Directory Data Group binds it together with the ITSO Shell Environment and any IPE and Value Record Data Group present on the CM.

### 2.4.4 IPE Data Groups

IPE Data Structures are bound into IPE Data Groups that identify the IPE Owner and have a Seal to provide for strong data authentication. The Seals are also bound to the ITSO Shell and Directory Data Groups. IPE Data Groups shall contain data that is either fixed for the life of the IPE or is rarely changed.

**2.4.5 Orphan IPE Data Groups**

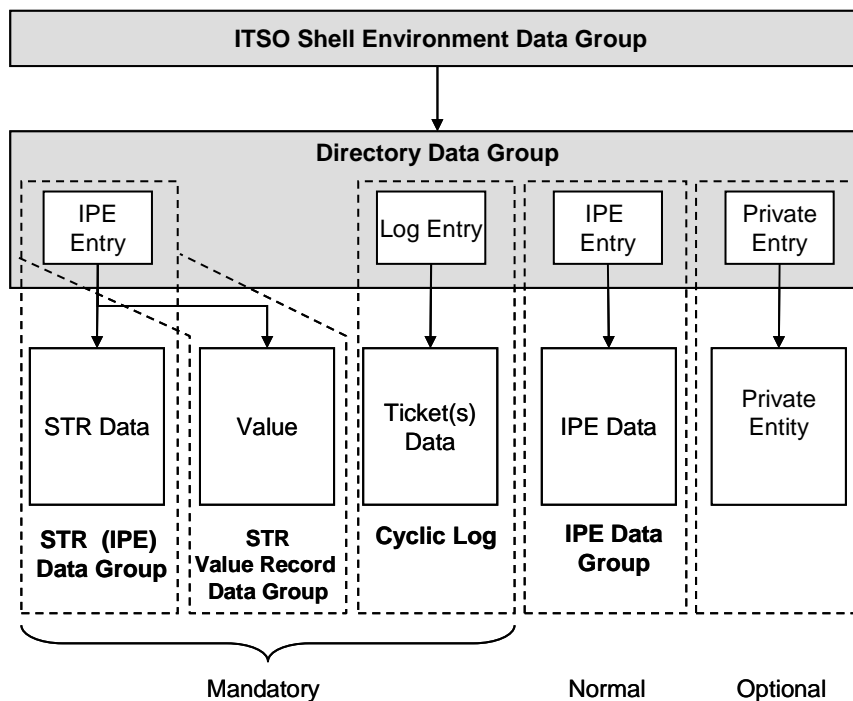
These Data Groups, if present, use IPE Data Group Structures but are not bound to the ITSO Shell and Directory Data Groups. The Orphan IPE Data Group shall be used with Single IPE ITSO Shells and in the Cyclic Log as required.

**2.4.6 Value Record Data Groups**

Where frequently changing data objects form part of an IPE, then a Value Record Data Group shall be associated with and bound to the IPE Data Group. Changes to Value Record Data Groups shall incorporate Anti-tear protection.

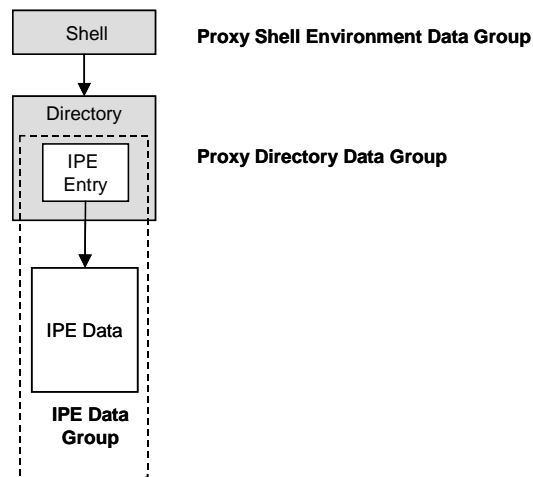
**2.4.7 General requirements**

The ITSO Shell shall support the ITSO Shell Environment Data Group, the Directory Data Group, Cyclic Log, and a Stored Travel Rights (STR) IPE Data Group. Provision (i.e. space) for at least one more IPE Data Group shall also normally be provided. The inclusion of private entries is optional but shall not be of such size as to preclude the inclusion of the aforementioned normal IPE Data Group. Figure 4 illustrates this.



**Figure 4 - Relationship between ITSO Shell Environment, Directory, IPE Data Groups and Cyclic Log**

In addition support is provided for CM with limited memory capacity. In this case the ITSO Shell and Directory Data Groups may be reduced to a small number of Data Elements that shall be associated with an Orphan IPE Data Group to form a Single IPE ITSO Shell. For processing by the ITSO Security Application Module (ISAM) the reduced ITSO Shell and Directory Data Groups shall be expanded by the Point of Service Terminal (POST) as defined in the CMD. Figure 5 illustrates a single IPE ITSO Shell.



**Figure 5 - Single IPE ITSO Shell**

All Cyclic Redundancy Checksums (CRCs) in this Part of ITSO TS 1000 shall be calculated in accordance with ISO/IEC 13239; an example is given in Annex A.

**2.4.8 The Cyclic Log**

Space within the ITSO Shell shall be assigned for the logging of tickets or other types of event. This takes the form of a Cyclic Log of two records, where the latest record created is pointed to by the Log Entry in the Directory. Each record shall be updated in chronological order. After the second record has been written the next record overwrites the first record. The Log is tear resistant in so far as updates do not affect a history of 1 previously valid record. Each Record of the Log shall occupy a single Sector, or, where the sector is large enough, an offset within a sector on the CM.

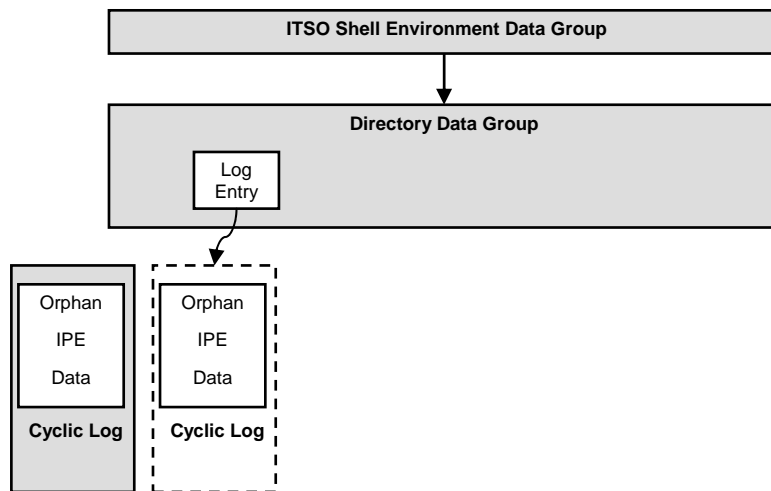
When a Cyclic Log is created without any records then the PTR Data Element of the Log Directory Entry (see clause 8.1.3) shall be set to a value of 0 and it is recommended that both prospective records shall be unsealed and populated with all 0's.

The initial Sector in the Log shall be the Starting Sector available for Logs in the CM (i.e. ST(i) where i = e#) and shall hold record T0 of the Cyclic Log. The next used Sector shall hold record T1 and shall have its related SCT(i) value set to 0. This differs from that used for IPE or Value Record Data Groups (see clause 5.1.5.5. for details).

After a Log record has been written the Log Entry in the directory shall subsequently be updated to point to the latest record created.

The data content of the records are as defined in the Transient Ticket Record Data Definition (see ITSO TS 1000-5) which shall be encapsulated as the Dataset of an Orphan IPE Data Group.

Figure 6 illustrates the Cyclic Log of 2 records.



**Figure 6 - Cyclic Log of two records**

Update rules:

1. Initially assume that the Log Entry in copy A of the directory indicates record T1 of the Log and is the latest record.
2. Record T0 shall be the next record to be overwritten.
3. Once record T0 has been overwritten, the Log Entry read from copy A of the directory Log Entry is modified to indicate record T0 is now the latest record and replaces the entire Log Entry in copy B of the directory.
4. In the event that the Directory Data Group is updated without a Log record having been created then the Log entry and related Sector Chain Table entries shall be copied in their entirety from the current Directory Data Group to the updated version of the Directory Data Group.

Creation / Removal rules:

The Log Directory Entry shall be created at the time the Shell is created and once created shall not be removed.

### 3. The Customer Media architecture

ITSO specifies a variety of CM that can hold an ITSO Shell. All ITSO CM share a common architecture defined in this clause. However there are physical differences between CM, which vary from simple memory cards to complex microprocessor based cards and other devices. These differences are catered for in the Customer Media Definitions (CMDs) found in ITSO TS 1000- 10.

An ITSO CM shall have the ability to store Data Groups within Sectors<sup>2</sup> of memory on the CM. The location of the ITSO Shell Environment and Directory Data Group(s) shall be fixed for the life of the ITSO Shell. Sector sizes and locations for a given CM are defined in the ITSO TS 1000-10 against a given CMD.

IPE and Value Record Data Groups may be added or removed throughout the life of the ITSO Shell and may occupy multiple memory Sectors that need not be contiguous. The SCT in the Directory Data Group defines the Sector(s) that any given IPE Data Group currently occupies.

Note: By this means, CM designers can optimise their particular memory architecture to best suit the types of IPEs they are likely to encounter, whilst at the same time, by allowing IPE Data groups to occupy non contiguous Sectors, maximise the reuse of the fragmented physical memory space that accumulates over time when IPE Data Groups of differing lengths have been added and removed from the ITSO Shell.

#### 3.1 CM electrical and physical characteristics

Note: The interface between the CM and the POST as defined in ITSO TS 1000-3 complies with all parts of ISO/IEC 14443. Cards of the Type A derivative, known under the trademark Mifare™ Classic are also supported.<sup>3</sup> The CM supports those parts of ISO/IEC 14443 as defined in ITSO TS 1000-10 for a given CMD.

Full details of the front and rear graphics design do not form part of this Specification. However where a "single application" CM containing only the ITSO Shell is issued by an ITSO Licensed Member the holder of said CM shall at least be able to readily view the ISRN without the need to use another device.

In the case where the CM is a card, it is recommended that the ISRN should be indelibly printed on its front or rear face. For ease of readability, the number may be printed in groups separated by double spaces. For example:

**633597●●7890●●1234●●5678** (where ● represents a character space).

Note: The first Block of six digits (633597) is the International Issuer Number registered to ITSO to facilitate international interoperability. The next Block of four digits (7890) would be set to the OID of the ITSO Shell Owner whilst the remaining digits are the unique ITSO Shell serial number and check digit...

#### 3.2 CM memory organisation

The ITSO Shell Environment Data Group shall occupy a defined location in the CM as specified in the CMD.

The Directory Data Group(s) shall occupy defined location(s) in the CM as specified in the CMD.

---

<sup>2</sup> The term Sector is used in a similar manner to its use in the context of a floppy disc or hard drive, where physical memory is subdivided into smaller manageable sectors for reasons of access speed and convenience of handling for files of an average size. In much the same way ITSO has prescribed the notion of Sectors on the wide variety of CM it intends to approve. In this context, a Sector may indeed be a sector on a memory card, an elementary file or record on a microprocessor card, or an alias on a java card or other device.

<sup>3</sup>This means that the CM coupling device shall not only support CM of both Types A and B, it must also support the Mifare™ Classic security architecture and protocols which differ from those defined in ISO/IEC14443 Part 4 for type A. Additional interfaces standards may also be incorporated where appropriate, so long as they do not interfere physically or electrically with CM defined in the ITSO Specifications.

IPE, Value Record and the Cyclic Log shall occupy those memory Sectors on the CM as interpreted from the SCT in accordance with the relevant CMD.

Value Record and Directory Data Groups may be duplicated on the CM depending on the method of Anti-tear protection employed.

### 3.3 Anti-tear protection

When using the contactless interface there is a risk that the CM may have been removed from its energising field during the writing of data to the CM. This could lead to corruption of data on the CM. ITSO specifies that any data carried on the CM that is changed and is critical to the correct operation of an ITSO Compliant Scheme shall be protected against accidental corruption. Methodology to ensure this can be achieved is commonly known as Anti-tear protection.

The contents of the following Data Groups shall be defined as critical data:

- the Directory Data Group;
- IPE Data Groups;
- Value Record Data Groups;
- the Cyclic Log.

When creating or altering the contents of any critical data it shall be ensured that:

Either:

All intentional data alteration shall be accomplished completely, correctly and without changes to other CM data.

Or:

There shall be no apparent<sup>4</sup> change to the contents of the CM.

#### 3.3.1 Mechanisation of Anti-tear protection

In general the requirements of this clause can be achieved in a number of ways dependent on the CM and the POST environment where CM data content is changed. These options can be categorised as:

- software only Anti-tear;
- hardware Anti-tear;
- tear prevention.

The following sub-clauses elaborate on these categories

##### 3.3.1.1 Software only Anti-tear

This is the most generic form of Anti-tear protection and may be used on any ITSO CM and shall be supported in any POST environment.

Critical data that is subject to frequent change during the life of an IPE shall be held in the Value Record Data Group associated with that IPE. When software only Anti-tear is used<sup>5</sup>, sufficient memory space shall be available

---

<sup>4</sup> Some data in the card may well have been changed correctly but other data may have been corrupted. In this event any POST shall be able to subsequently access the previous fully functional state of the card.

<sup>5</sup> Full details of the logic involved in the implementation of software Anti tear applicable to each CMD are to be found in TS 1000 Part 10

on the CM to contain: two copies of all Value Record Data Groups that may be updated in any one Transaction and two copies of the Directory Data Group.

The copies, conceptually known as the A copy and the B copy respectively; alternately contain the latest and previous versions of the contents of the Data Group. Both copies shall be stored on the CM in such a manner that if one version (normally the latest) is corrupted the other version shall still be usable.

The physical locations where the A & B copies may be stored depend on the memory technology used by the CM. For example, when using EEPROM technology where erase / re-write cycles may only be achieved on a page-by-page basis, copy A shall reside in a different page of memory to its associated copy B.

The SCT in the Directory Data Group points to which copy of any of the Value Record Data Groups contain data that shall be deemed to be the latest version. The Directory Data Group shall always be the last Data Group altered after any number of Value Record Data Groups has been altered.

The Directory Data Group contains a sequence number where the highest number shall indicate the latest version of the Directory Data Group.

Note: Once the Directory is successfully modified it will commit the CM to its new state and will point to the latest versions of all the Value Record Groups and Cyclic Log records. If the Directory is not successfully modified then the other copy of the Directory will point to the previous versions and the CM will appear not to have been changed at all.

Software only Anti-tear shall be used where the CM does not support adequate hardware Anti-tear.

The method of implementation of software only Anti-tear for any given CM shall be as defined in the CMD.

### **3.3.1.2 Hardware Anti-tear**

Some CM may have in-built Anti-tear protection systems that are a combination of CM firmware and hardware, ITSO does not preclude the use of hardware Anti-tear as a replacement for software only Anti-tear methods, providing the requirements of clause 3.3 are met.

The use of hardware Anti-tear shall be as indicated in the CMD.

### **3.3.1.3 Tear prevention**

Some operations such as loading a new IPE should be carried out in an environment where there can be much greater control over the CM whilst it is being altered. In such cases, the CM may be given to the loading agent or placed in a special receptacle where accidental removal is minimised.

In general, if loading takes place where:

- the POST is manually attended;
- the CM is held clamped by the POST;
- access to the service is physically prevented until the CM is known to be altered correctly;
- an alternative method is used that has been accredited by ITSO.

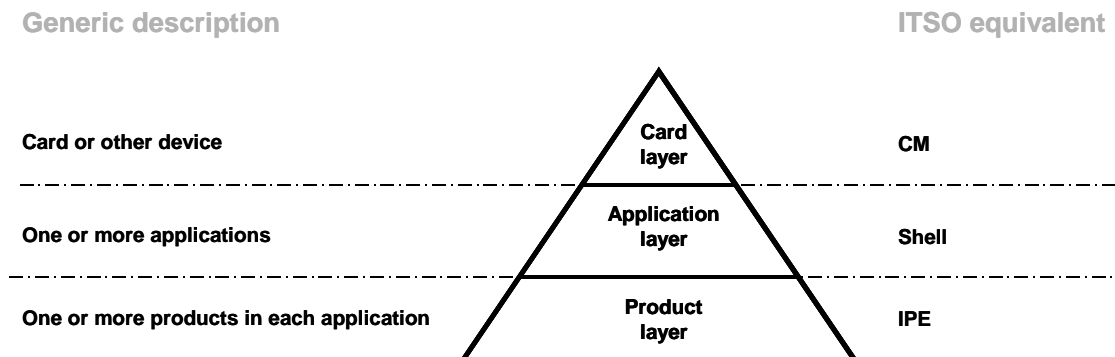
The Mediaholder can be advised to re-present the CM for re-writing in the event that data is known to be incorrect. In these cases the application software shall ensure that the same instance of the same CM is being re-presented for correction.

Tear prevention methods shall be applied to any creation or modification of an IPE Data Group.

Tear prevention methods shall be applied to all Data Group and CM access key loading operations when creating an ITSO Shell.

### 3.4 Relationships between CM, ITSO Shell, applications and IPEs

An ITSO Shell containing many IPEs shall reside on a variety of CM. The CM may be exclusively used by ITSO or used in conjunction with other applications. Thus a generic multi-application CM structure shall have up to three layers as illustrated in Figure 7.

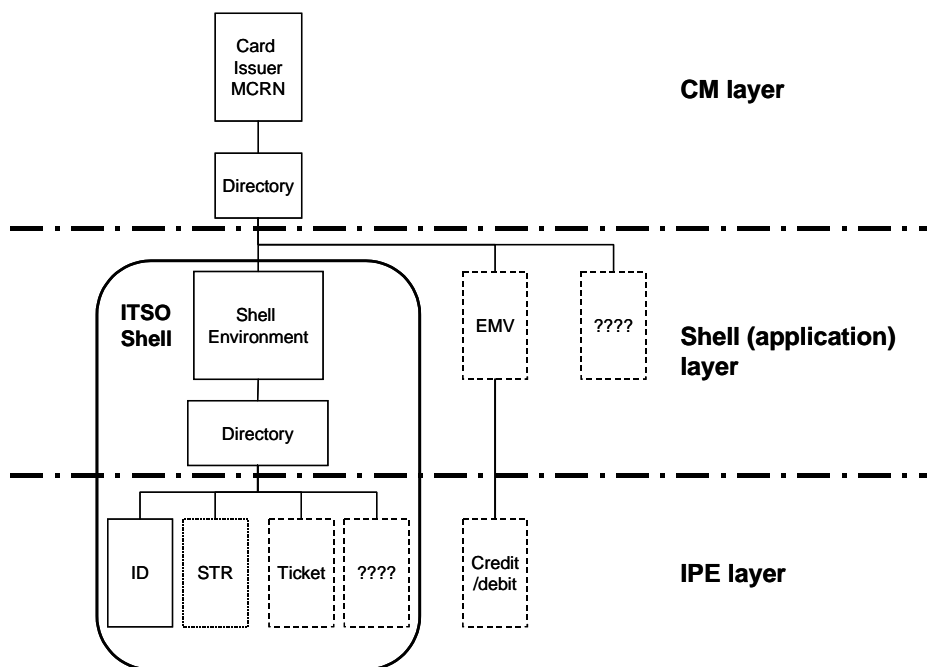


**Figure 7 - Generic multi-application CM structure**

In this structure one or more Products (IPEs) may be related to a single instance of a particular application (ITSO Shell) and one or more applications are in turn related to a single instance of a card (CM).

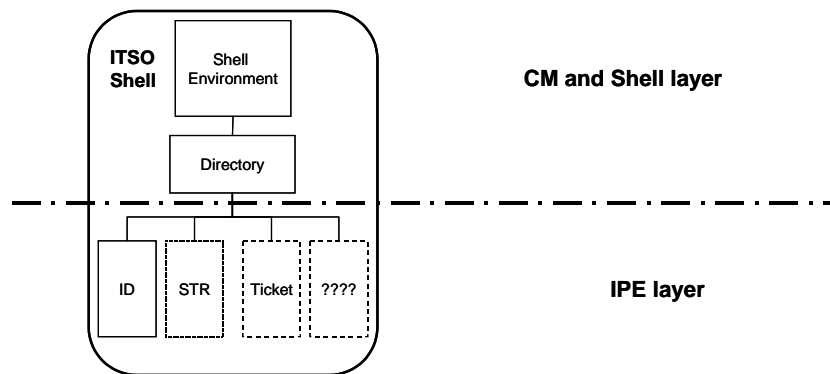
The ITSO Shell carries an internationally registered reference number, a Directory of the IPEs it holds and the IPEs themselves. This is sufficient information to allow the ITSO Shell to be uniquely identified alongside other applications that may in turn be held on CM belonging to another scheme.

The ITSO Shell and its relationship within the generic multi-application CM structure are illustrated in Figure 8.



**Figure 8 - ITSO Shell on a multi-application CM**

Alternatively, since the ITSO Shell combines the attributes of both CM and application layer it can be used as an ITSO CM in its own right as shown in Figure 9.



**Figure 9 - ITSO Shell as the CM**

**3.5 Access to the Shell**

There are a number of issues to be addressed when accessing the ITSO Shell. These are:

- Selecting the ITSO Shell.
- Determining the CM.
- Accessing the CM layer number.
- Security.

**3.5.1 Selecting the ITSO Shell**

ITSO requires the transport application to be accessed and activated as quickly as possible after a CM is energised by the contactless interface. If a multi application CM is used it is recommended that the ITSO Shell be implicitly selected when the CM is energised from the contactless interface.

This does not preclude the presence of other applications accessed via a contactless or contact interface, on the same CM. In this event there are further methods to select a contactless application available to the application software running in the POST.

**3.5.1.1 Use of an ISO Application Identifier (AID)**

The ITSO AID, where supported, may be used to access an ITSO application directly from a contact interface or to return to the ITSO application via the contactless interface from another contactless application.

In accordance with ISO/IEC 7816-5, the AID shall be made up of:

- registered Application Provider Identifier (RID) 5 bytes;
- proprietary Application Identifier Extension (PIX) 6 bytes.

The international RID assigned to ITSO is (in hex): A0, 00, 00, 02, 16

The registration category for this RID is ‘international’ and as such is represented by the A (hex) in the 4 most significant bits.

The 6 Byte PIX field shall represent the label “ITSO-1” in ASCII and is (in hex): 49, 54, 53, 4F, 2D, 31 All other values are Reserved for Future Use (RFU) by ITSO.

Thus the complete ITSO AID shall be (in hex) A0, 00,00,02,16, 49, 54, 53, 4F, 2D, 31.

### 3.5.1.2 Use of an Application Family Identifier (AFI)

This technique is described in detail in ISO / IEC 14443 whereby during the process that selects a contactless card from others that may be present, specific families of contactless applications can be included in the search (i.e. others will be ignored).

A CM may optionally support this feature and codes used in this process have so far been defined for Transport, Financial, Identification, Telecommunications, Medical, Multimedia, Gaming, and Data storage.

Codes also exist for selecting groups of application families including a code for "all applications".

If this mechanism is present and is used to override the implicit selection recommended by ITSO, in a non-ITSO application, co resident applications need not suffer any degradation of performance.

### 3.5.2 Determining the CM

There is a weakness in both the contact and contactless international standards regarding the lack of definition of a uniquely identifiable mechanism that allows a POST to easily determine the hardware and software build standard of the CM with which it intends to communicate.

The ITSO Specification uses three bytes to define the CM in terms of:

- the basic build standard - defined by the Format Version Code (FVC);
- the basic security architecture - defined by the Key Strategy Code (KSC);
- the version of the key set carried - defined by the Key set Version Code (KVC).

These bytes are stored in the ITSO Shell Environment Data Group but, since the FVC also defines the location and structure of the ITSO Shell Environment Data Group, this presents a challenge to an ITSO Compliant Scheme designer.

ITSO addresses the problem by explicitly specifying the location of the ITSO Shell environment in the context of secure memory cards and generic microprocessor cards and makes them freely readable. It leaves it up to the POST to determine which type of CM it has encountered. Whilst this is a trivial issue when, as currently, only a few very different CM have been defined, it becomes less satisfactory when more CM are included.

Current secured memory card technologies do not offer this option and can only be accessed by using non-deterministic methods.

Simple memory cards that have less access restriction can utilise the same data object structure as for microprocessor cards and again this gives us a solution that enables support of any new simple memory card platforms that ITSO may wish to include in the future.

### 3.5.3 Accessing the CM layer number

Every ITSO Shell has a unique ISRN and in the case of an ITSO-only CM this shall be the CM number.

Where the ITSO Shell is present on a multi-application card, the card as a whole is likely to be referenced by an alternative method. Whilst use of the ITSO Shell in any ITSO Compliant Scheme stands alone and does not require reference to any other card numbers, there may be occasions when access to the Multi application Card Reference Number (MCRN) held in the card layer is desirable. Access to the MCRN may be required when an ITSO Shell is added to an existing card or when a card is lost or stolen and it might be desirable to try and block the whole card.

Where the MCRN is only present in the card layer then it may, dependent on card platform, be extremely complex and time consuming to access from the contactless interface. To simplify the situation ITSO has specified an optional Data Element within the ITSO Shell Environment Data Group that holds a copy of the MCRN.

### 3.5.4 Security

Access to the ITSO Shell is CM dependent, whereas the IPE Owner determines IPE data security independent of the CM.

ITSO deals separately with:

- access to the data held in an ITSO Shell;
- the cryptographic Seal applied to ITSO IPEs.

For a given CM, the KSC and KVC codes, held in the ITSO Shell Environment Data Group, define the cryptographic processes to be used when accessing the ITSO Shell. These processes are defined in ITSO TS 1000- 7 and ITSO TS 1000-8.

For a given IPE Data Group and Value Record Data Group the cryptographic processes to be used are determined by the ITSO ISAM from the group label, or in the case of the Directory Data Group implied from the ITSO Shell Environment Data Group. These processes are defined in ITSO 1000 Parts 7 & 8.

### 4. The ITSO Shell Environment Data Group

The ITSO Shell Environment Data Group consists of only a single Data Structure illustrated in Figure 10. There is no Seal for this structure; a CRC at the end of the Dataset covers all preceding Data Elements. Once created the ITSO Shell Environment Data Group's contents shall be set to be "read only" for the life of the ITSO Shell.



Figure 10 - The ITSO Shell Environment Data Group

#### 4.1 ITSO Shell Environment Data Group Dataset

The contents of this Dataset are shown in Table 1. The rest of this clause describes each Data Element in turn.

Table 1 - ITSO Shell Environment Dataset Data Elements

Byte offset #	# of bits	Data type	Label	Description of Data Element
0	6	BIN	ShellLength	The length of the ITSO Shell Environment Dataset (in Blocks)
0.75	6	BMP	ShellBitMap	The ITSO Shell Environment Bit-Map
1.5	4	HEX	ShellFormatRevision	Defines the Format Revision number of this ITSO Shell.
2	24	IIN	IIN	Issuer Identification Number (registered to ITSO)
5	16	BCD	OID	ITSO Shell Owner Operator Identification Number
7	28	BCD	ISSN	ITSO Shell Serial Number
10.5	4	BCD	CHD	Check Digit <sup>6</sup>
11	8	HEX	FVC	Format Version Code
12	8	HEX	KSC	Key Strategy Code
13	8	HEX	KVC	Key-set Version Code
14	2	BIN	RFU	Reserved for Future Use by ITSO
14.25	14	DAT E	EXP	ITSO Shell EXPiry date
16	8	HEX	B	Size of memory Sector
17	8	HEX	S	Maximum number of Sectors supported
18	8	HEX	e#	Maximum number of Directory Entries supported
19	8	HEX	SCTL	Size in bytes of the Sector Chain Table
20	80	BCD	MCRN	Optional: Multi-application Card Reference Number
AR	AR	PAD	Padding	Padding with zeros as required to ensure the entire Dataset is a whole number of Blocks in length
22 or 30	16	HEX	SECRC	ITSO Shell Environment Cyclic Redundancy Check (SECRC) covering the preceding Data Elements in this Dataset

<sup>6</sup> This digit is a check digit calculated in accordance with ISO/IEC 7812-1

Note: Commonly used Data Structures are colour-coded throughout this document as an aid to recognition.

**4.1.1 ShellLength**

This is a 6 bit binary integer, the value of which shall specify the entire length (inclusive of this element, any padding element and the final SECRC) of the ITSO Shell Environment Dataset in Blocks of BL bytes, where BL is specified in accordance with the Value of the ShellFormatRevision defined in Table 2

**Table 2 - Coding of BL**

Value of ShellFormatRevision	Value of BL	Length of Dataset
0x1	4	ShellLength x 4
All other values	RFU	ShellLength x BL

**4.1.2 ShellBitMap**

This is a set of six flag bits that hold information about the data that is present in the ITSO Shell Environment Dataset. The flag bits for specific conditions are coded as shown in Table 3, where x indicates that the setting of that flag is irrelevant to that particular condition. Only the Least Significant 2 Bits of the ShellBitMap are currently defined. The remaining flags in the set shall be set to zero and RFU by ITSO.

**Table 3 - Shell-Bit-Map flag usage**

Byte Offset	0		1			
	1	0	7	6	5	4
<b>Condition</b>	<b>MSB</b>					<b>LSB</b>
Compact ITSO Shell	0	0	0	0	0	0
Full ITSO Shell	X	X	X	X	X	1
MCRN not present	X	X	X	X	0	X
MCRN present	X	X	X	X	1	X
RFU	X	X	X	X	-	-

**4.1.3 ShellFormatRevision**

This is a 4 bit binary integer, the value of which, in the range 1 – F (hex) shall indicate the revision number for this version of the ITSO Shell Environment Data Group.

**4.1.4 ITSO Shell Reference Number (ISRN)**

The ISRN is the concatenation of the following 4 Data Elements that shall uniquely identify this instance of the ITSO Shell.

**4.1.4.1 ITSO Issuer Identification Number (IIN)**

This is a six BCD digit number registered with ISO as a unique global identifier.

Note: ITSO has been assigned an IIN of 633597 by ISO.

**4.1.4.2 ITSO Operators Identification Number (OID)**

In this context this is a four BCD digit number, in the range 0001 to 8,000, that ITSO registers and allocates to a Licensed Member who is responsible for (owns) the ITSO Shell. The ITSO Shell may be issued as a CM in its own right or added onto another CM. Numbers outside the range stated are RFU by ITSO.

#### 4.1.4.3 ITSO Shell Serial Number (ISSN)

This is a seven BCD digit number, in the range 0 – 9,999,999, that Licensed Members shall use to ensure their ITSO Shells are uniquely identified.

#### 4.1.4.4 Check Digit (CHD)

This is a single BCD digit, which is calculated to be a check digit for all the preceding digits of the ISSN. The check digit shall be calculated using the Luhn formula for computing modulus 10 "double-add-double" check digit as defined in ISO/IEC 7812-1. Verifying this check digit is recommended whenever customers quote their ISSN.

#### 4.1.5 Format Version Code (FVC)

This is a single byte binary integer in the range 1 – 255, the value of which shall identify the format version of the CM. The FVC indicates which CMD is used. The value 00 (hex) is RFU by ITSO.

#### 4.1.6 Key-Strategy Code (KSC)

This is a single byte binary integer in the range 1 – 255, the value of which is subservient to the Format Version Code and shall identify the key strategy to use. The value 00 (hex) is RFU by ITSO.

The KSC indicates which security algorithm shall be used to access data groups on a CM of FVC (n), (see ITSO TS 1000-7 and ITSO TS 1000-8).

#### 4.1.7 Key-set Version Code (KVC)

This is a single byte binary integer in the range 1 – 255, the value of which is subservient to the Key Strategy Code and shall identify which version of key-set to use. The value 00 (hex) is RFU by ITSO.

The KVC indicates which key-set is used to access to data groups on a CM of FVC (n) having KSC (m), (see ITSO TS 1000-s 7 and 8).

#### 4.1.8 ITSO Shell EXPIRY date (EXP)

This is the date, normally the end of a month, after which, the ITSO Shell shall no longer be valid in normal use.

Coded as a Data type DATE, this Data Element may be set to all zeros indicating:

- no expiry if the ITSO Shell is alone on an ITSO CM;
- the expiry date is devolved to that of the multi-application CM upon which the ITSO Shell resides.

Once set, changes to the ITSO Shell Expiry date Data Element are not permitted during the life of the ITSO Shell.

Note: If the multi-application CM has an expiry date, then it is recommended that a date no later than the multi-application CM expiry date shall be copied across to this Data Element when the ITSO Shell is installed.

#### 4.1.9 Size of memory Sector (B)

This is a single byte binary integer in the range 1 – 255 as defined in the CMD. The value 00 (hex) is RFU by ITSO.

#### 4.1.10 Number of Sectors (S)

This is a single byte binary integer in range 1 – 255 that shall indicate the maximum number of Sectors of memory available in this ITSO Shell for use by All Data Groups and the Cyclic Log. The value 00 (hex) is RFU by ITSO.

Note: The actual number of Sectors used by a particular CM for Data Groups and the Cyclic Log is determined from the contents of the Sector Chain Table found in the Directory Data Group.

#### 4.1.11 Maximum number of Directory Entries (e#)

This is a single byte binary integer in the range 1 - 31 that shall indicate the maximum number of entries in the Directory of this ITSO Shell. The memory space occupied by these entries is pre-allocated within the Directory Data Group. This space may be calculated as (e# x the size of a Directory Entry) bytes, which since a Directory Entry is 5 bytes becomes 5(e#) bytes. The value 00 (hex) and values in the range 32 – 255 are RFU by ITSO.

#### 4.1.12 Size in bytes of the Sector Chain Table (SCTL)

This is a single byte binary integer in the range 1 – 255 that shall indicate the length, in bytes, of the list of Sector Data Elements (SCT(i)) held by the Sector Chain Table in the Directory Data Group. The value 00 (hex) is RFU by ITSO.

#### 4.1.13 Optional: Multi-application CM reference (MCRN)

This optional Data Element shall, if included, hold a duplicate copy of the CM reference number as defined by the issuer of a multi-application CM.

The Data Element is to be coded in BCD including a check digit in accordance with ISO/IEC 7812-1.

The MCRN may be up to 19 digits long according to ISO/IEC 7812-1. It shall be terminated with at least one F(hex) character and suffixed with additional F(hex) characters as required to ensure this Data Element has a fixed length of 10 bytes.

#### 4.1.14 Padding

A number of binary zeros inserted, as required, to pad the entire Dataset to a whole number of Blocks. The size of the Dataset in bytes is given by ShellLength x BL where BL is defined in Table 2 of this clause.

#### 4.1.15 ITSO Shell Environment Checksum (SECRC)

This is a two byte binary integer containing a cyclic redundancy check calculated over all the preceding Data Elements in the ITSO Shell Environment Data Group. This shall be used to check the data integrity of the Data Group. The SECRC is calculated in accordance with the method described in Annex A of this document.

### 4.2 Compact ITSO Shell Environment Dataset

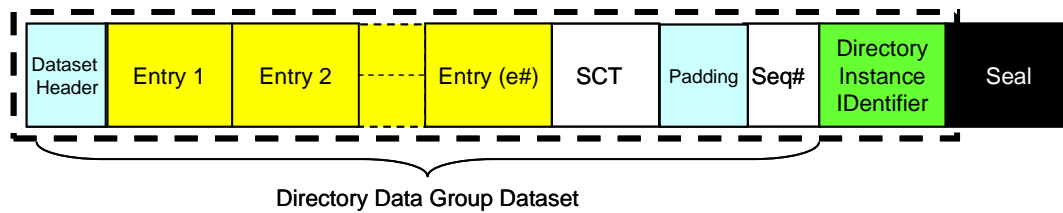
Where the CM has insufficient memory capacity to support a full ITSO Shell, a compact version of the ITSO Shell consisting of only the data elements shown in Table 4 shall be used. This compact ITSO Shell shall be expanded by the POST application to replicate a full ITSO Shell as defined in the CMD as indicated by the value of the FVC data element.

**Table 4 - Compact ITSO Shell**

Byte offset #	# of bits	Data type	Label	Description of Data Element
0	6	BIN	ShellLength	The length of the ITSO Shell Environment Dataset (in Blocks)
0.75	6	BMP	ShellBitMap	The ITSO Shell Environment Bit-Map
1.5	4	HEX	ShellFormatRevision	Defines the Format Revision number of this ITSO Shell.
2	8	HEX	FVC	Format Version Code

## 5 The Directory Data Group

The Directory Data Group consists of three structures, the Dataset (which includes Directory Entries for IPEs, Log records and Private Applications); the Directory InstanceID; and the Seal. This is illustrated in Figure 11.



**Figure 11 - Construction of the Directory Data Group**

The Directory Data Group shall have enough reserved space to hold the maximum number of Directory Entries(E) specified for the particular CM it is on. Unused Directory Entries shall be set to all zeros. For some IPEs the Directory Data Group information alone may be sufficient to allow a Transaction to proceed without accessing an associated IPE Data Group.

### 5.1 Directory Data Group Dataset

The contents of the Directory Data Group’s Dataset are shown in Table 5. Each Data Element is then described in turn.

**Table 5 - Directory Data Group Dataset Data Elements**

Byte offset #	# of bits	Data type	Label	Description of Data Element
0	6	BIN	DIRLength	RFU
0.75	6	BMP	DIRBitMap	The directory bit map
1.5	4	HEX	DIRFormat Revision	The Format Revision number of this Directory Data Group
2	40 x e#	HEX	E(i)	Directory Entries of 5 bytes each for (i = 1 to e#) Set to all zeros if not used
2+5(e#)	$\Psi \times S-3$	BIN	S-3CT(i)	An array of Sector numbers of $\Psi$ bits each for (i = 1 to S) which associates the Directory Entries to the location(s) where the Data Groups are stored. Known collectively as the Sector Chain Table (SCT). The SCT has a length of SCTL bytes
2+5(e#) + SCTL	8	HEX	DIRS#	The sequence number of this copy of the Directory Data Group

#### 5.1.1 DIRLength

This 6 bit Data Element is RFU by ITSO.

Note: In this version of the Specification the Directory Data Group Dataset Length shall be calculated, to a precision of 1 Byte, from the e# and SCTL Elements present in the ITSO Shell Environment Data Group. This allows optimisation of the Directory Data Group to suit any chosen Sector size.

**5.1.2 DIRBitMap**

This is a set of six flag bits that shall indicate which Directory Entries are Log entries and whether the entire ITSO Shell has been blocked. The coding of two of the flags determines the use of the last two Directory Entries (i.e. where  $E(i) = E(e\# - 1)$  and  $E(i) = E(e\#)$  respectively) such that they may be entries for; two IPEs, or a single IPE and one Log entry, or two Log entries. The flag bits for all conditions are coded as shown in Table 6, where X indicates that the setting of that flag is irrelevant to that particular condition.

**Table 6 - DIR Bit Map**

Byte offset	0		1			
Bit position	1	0	7	6	5	4
Condition	MSB					LSB
ITSO Shell not blocked	X	X	X	X	X	0
ITSO Shell blocked	X	X	X	X	X	1
No Log entries present	X	X	X	0	0	X
The last Directory Entry is a Log entry	X	X	X	0	1	X
RFU (See note below)	X	X	X	1	0	X
RFU	X	X	X	1	1	X

Only the Least Significant 3 Bits of the DIRBitMap are currently defined. The remaining flags shall be set to zero (and are RFU by ITSO).

Note: The DIR Bit Map coding shown in the shaded row in the above table may be encountered on certain compatible cards that have been coded in compliance with earlier versions of the ITSO specification (with two log entries occurring in the last two directory entries).

In this case:

Either: Treat the code 10 as code 01 and ignore any penultimate Directory Entry Log.

Or: As an option bring the CM into compliance with this version of the specification.

For further details see ITSO DG0010.

**5.1.3 DIRFormatRevision**

This is a 4 bit binary integer in the range 1 – 15 that shall indicate the revision number for this version of the Directory Data Group. This value is defined for each CMD in ITSO TS 1000-10. The value 0 (hex) is RFU by ITSO.

**5.1.4 Directory Entries**

Directory Entries shall be associated with IPE Data Groups or the Log.

**5.1.4.1 IPE Directory Entry**

Each IPE Data Group or Private Application present in the ITSO Shell shall have an entry in the directory. The IPE Directory Entry is identical to the label for the IPE Data Group itself and indicates: the owner of the IPE, the type and sub type of IPE and its expiry date. The IPE Directory Entry is bound cryptographically to both the Directory and the IPE Data Groups. For some applications the IPE Directory Entry information alone may be sufficient to

allow a transaction to proceed without accessing the associated IPE Data Group. The contents of the IPE Directory Entry are defined in detail in clause 6.1.

#### 5.1.4.2 Log Directory Entry

Any Log present in the ITSO Shell shall have a Log Directory Entry that is bound cryptographically to the Directory Data Group. The Log Directory Entry is used as the Directory Entry associated with the Cyclic Log. It points to the latest record created and may also be used, to provide entry/exit and pass-back indication that is independent of the latest record created. The contents of the Log Directory Entry are defined in detail in clause 8.

#### 5.1.5 Sector Chain Table array (SCT)

A variable length array of maximum dimension ( $\Psi \times S-3$ ) bits shall store a logical map of the location of Data Groups, chained across Sectors where necessary, that is independent of the CM chosen. From a given start each Data Element in the array shall point to the next Sector occupied by a Data Group or collection of Data Groups. The coding of the last Element in the chain shall indicate the end of the chain and also shall contain information relating to the status of the related Data Group(s).

The relationship between the Sector numbers (SCT(i)) and the physical sectors (or files) where the Data Groups may be stored are defined in the CMD for each particular CM. For the purpose of accessing the ITSO Shell Environment Data Group it shall have a Sector number of 0. Likewise, Copy A of the Directory Data Group shall have a Sector number of (S-2) and Copy B a Sector number of (S-1). Thus the maximum number of Sectors available for IPE Data Groups and the Cyclic Log is always (S-3), hence the maximum number of Data Elements in the SCT is also (S-3).

The locations of the Directory and ITSO Shell Environment Data Groups shall be fixed and are not included in the SCT, but are defined in the CMD for each particular CM.

This array shall always be padded with zeros to the nearest whole number of bytes.

##### 5.1.5.1 Calculation of $\Psi$

$\Psi$  shall be calculated as the number bits it takes to code the maximum number of Sectors (S), in accordance with the formula  $S \leq 2^\Psi < 2S$ , i.e.:

for S = 16 then  $\Psi = 4$  and the maximum number of SCT Data Elements is 13

for S = 24 then  $\Psi = 5$  and the maximum number of SCT Data Elements is 21

Table 7 shows an example of an SCT where S = 16, the number of Data Elements in the SCT is 13, and  $\Psi = 4$

**Table 7 - Example coding of the Sector Chain Table array**

Byte offset #	Bit Position	Data type	Label SCT(i)
0	7,6,5,4	BIN	SCT1
0	3,2,1,0	BIN	SCT2
1	7,6,5,4	BIN	SCT3
1	3,2,1,0	BIN	SCT4
2	7,6,5,4	BIN	SCT5
6	7,6,5,4	BIN	SCT13
6	3,2,1,0	PAD	PAD

**5.1.5.2 Sector Chain Data Elements (SCT(i)) relating to IPE and Value Record Data Groups**

Each Data Element in the array normally contains the Sector number of the next Sector occupied by any given Data Group or collection of Data Groups.

There are a number of constraints to be taken into account when interpreting the array.

1. The first e# Sectors are normally reserved for the first Sector of each Data Group or collection of Data Groups having a matching Directory Entry position (see ITSO 1000 Part 10 for details). This identifies the start Sector relating to each Directory Entry.

i.e. SCT(1) shall be the first Sector of a Data Group or set of Groups pointed to by Directory Entry E(1);

SCT(2) shall be the first Sector of a Data Group or set of Groups pointed to by Directory Entry E(2);

and so on.

In this case the contents of the SCT(i)th Data Element shall be the value x where SCT(x) is the Sector that contains the next part of the Data Group or set of Data Groups associated with the same Directory Entry.

2. The next Sector occupied by the Data Group or set of Groups would normally be any Sector > e#. However, Sectors normally reserved for the first Sector of other groups may be utilised providing there is no valid Directory Entry in the matching position.
3. Sectors are chained together by reference from the previous SCT(i) until the last occupied Sector is encountered. In this case the content of SCT(i) shall:

- A) Refer to itself indicating the end of an IPE that has never been used

i.e. the contents of SCT(i) = (i);

- B) Refer to the Sector S-2 indicating the end of an IPE that is blocked

i.e. the contents of SCT(i) = (S-2);

- C) Refer to the Sector S-1 indicating the end of an IPE that has been used at least once and is not blocked

i.e. the contents of SCT(i) = (S-1).

4. A Value of SCT(i) = 0 indicates that the Sector does not contain any valid Data Group and is free for use.

**5.1.5.3 Ordering of Data Groups in the Sector Chain Table**

From the previous clause, the first Sector of a Data Group is referenced by the position of the Directory Entry in the Directory Data Group.

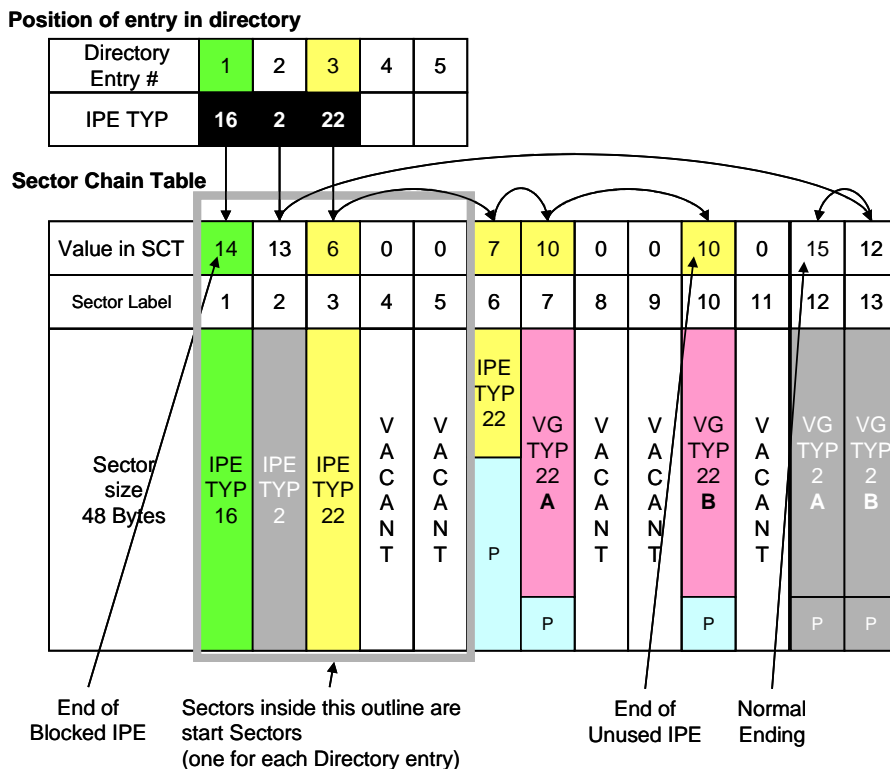
Given E(i), where (i) takes the value from 1 to e#, is the array of Directory Entries. Then E(1) is the first Directory Entry encountered in the Directory Data Group, E(2) the second, continuing in order until E(e#) which shall be the last.

E(i) may point to a single Data Group or a set of Data Groups maintained in order as follows:

1. For a single IPE Data Group the Data Group shall span as many Sectors as required in the order dictated by the Sector Chain Table (SCT) and shall finally be padded with Zero's to the end of the last Sector used by the Data Group.
2. For an IPE Data Group plus a Value Record Data Group the IPE follows 1 above then continues with the Value Record Data Group that shall start at the beginning of a new Sector pointed to by the next SCT entry, then again as in 1 etc, finally padded to the end of the last Sector used by the Value Record Data Group.
3. Where two copies of a Value Record Data Group are present, then copy A precedes copy B and copy B shall start at the beginning of the next Sector pointed to by the next SCT entry, then again following the rule as in 1 etc., finally padded to the end of the last Sector used by the group. When a Value Record is updated then the SCT is changed such that the current Value Record Data Group shall be encountered first in the chain. By this means the latest Value Records are to be found in the first Value Record Data Group encountered in the SCT.
4. In the case where an IPE consists only of two copies of a Value Record Data Group, the SCT(i) labels relating to both copies shall remain fixed at their initial conditions and an alternative method of determining the most recently modified Data Group is required.
5. For the Cyclic Log the SCT(i) labels relating to the sectors used by the Log shall remain fixed at their initial conditions.

**5.1.5.4 Example use of the Sector Chain Table for IPE and Value Record Data Groups**

Figure 12 illustrates how a single IPE (TYP 16) Data Group that is blocked, an IPE Data Group with two copies of a Value Record Data Group (TYP 22) that has not yet been used and an STR (TYP 2) that has been used may be chained within a CM that has 13 Sectors available for use, where B = 48, S =16, e# = 5.



**Figure 12 - Example of the relationship of Data Groups to the Sector Chain Table****5.1.5.5 Sector Chain Table Data Elements (SCT(i)) relating to the cyclic log**

The cyclic log contains records which are Transient Ticket Datasets enveloped as orphan IPEs. An orphan IPE shall be created each time a log record is written and in the case of Customer Media using Sector diversified access keys the ISAM will return access keys for any empty (i.e. where  $SCT(i) = 0$ ) Sectors.

Thus in order to overwrite a Sector within the cyclic log collection of records the next Sector scheduled to be overwritten shall have an SCT(i) value of 0. This prescribes a format for the Sector Chain Table that differs from that used for IPE or value Record Data Groups, as described in the succeeding paragraph and following examples.

**SCT Update rules when using the Cyclic Log**

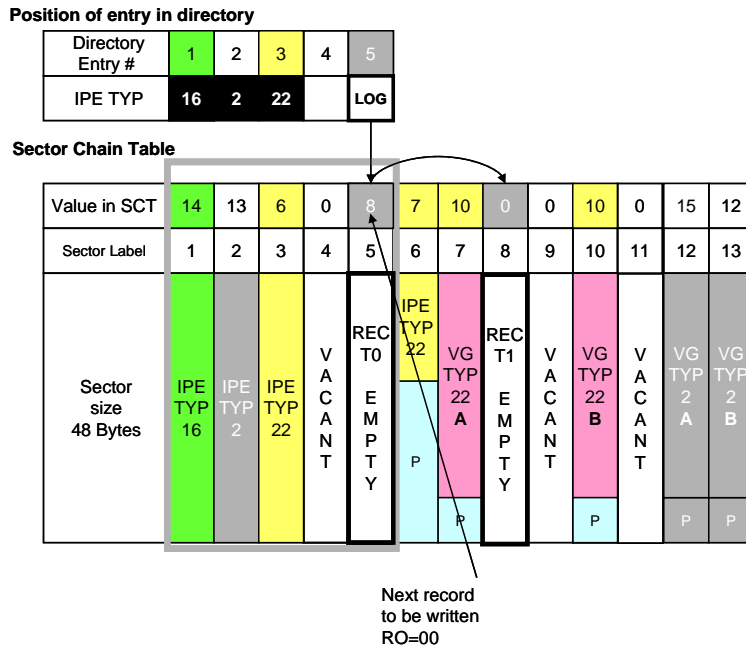
The Data Element in the SCT array relating to the start Sector of the Cyclic Log always contains the Sector number of the Sector occupied by the other record in the Log. In this case the SCT(i) for that Sector shall be set to 0. The latest record written is indicated by the setting of the Record Offset (RO) Data Element in the Log Entry, and is illustrated in clause 5.1.5.6. Thus the other record is the one to be written to next.

**5.1.5.6 Example use of the Sector Chain Table and directory entry for the Cyclic Log**

Figures 12a-c illustrates how a single Cyclic Log of two records may be chained within a CM that has 13 Sectors available for use, where  $B = 48$ ,  $S = 16$ ,  $e\# = 5$ .

a) Creation of Log

The Log may be created as empty if required by the CM owner. Figure 12a shows an empty Log of two records



added to the example given in figure 12.

Figure 12a – Cyclic Log as created and the Sector Chain Table

Table 12a The directory entry associated with the initial creation of an empty Log:

Byte offset #	Bit Position	Setting	Label	Description
0	7	bin1	LPF	Indicates Normal mode
0	6,5,4,3,2	bin00000	PTR	Pointer to the directory entry of the IPE most relevant to the last Log record created
0	1,0	bin00	EEI	Entry / Exit indicator
1	All	0xXXXXXX	DTS	Date Time Stamp as determined on creation
4	7,6	bin00	RO	Record Offset
4	5,4,3,2,1,0	bin000000	PTLBM	Passback Time relating to the use of IPE pointed to by PTR

The value of RO is set to bin 00 indicating that the Log may be empty and record T0 is the next record to be written

b) 1<sup>st</sup> Record (record T0) used

Figure 12b shows the SCT values after record T0 has been populated.

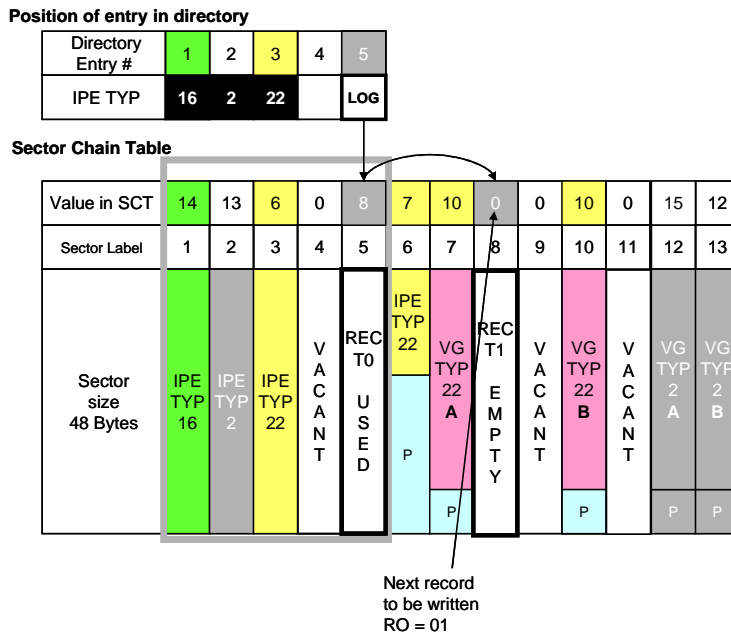


Figure 12b – Cyclic Log after writing to the record T0

(Note the Sector Chain Table does not change)

The example in table 12b assumes that the use of the TYP22 IPE caused the Log entry to be written and Passback Time processing is determined by the POST.

Table 12b The directory entry associated with the overwriting of record T0 in the Log:

Byte offset #	Bit Position	Setting	Label	Description
0	7	bin1	LPF	Indicates Normal mode
0	6,5,4,3,2	bin00011	PTR	Pointer to the directory entry of the IPE most relevant to the last Log record created
0	1,0	bin00	EEI	Entry / Exit indicator
1	All	0xxxxxxx	DTS	Date Time Stamp
4	7,6	bin01	RO	Record Offset
4	5,4,3,2,1,0	bin000000	PTLBM	Passback Time relating to the use of IPE pointed to by PTR

The value of R0 is set to bin 01 indicating that record T1 is the next record to be written. In this case Record T0 is the latest record.

c) 2<sup>nd</sup> record (record T1) used

Figure 12c shows the SCT values after record T1 has been populated.

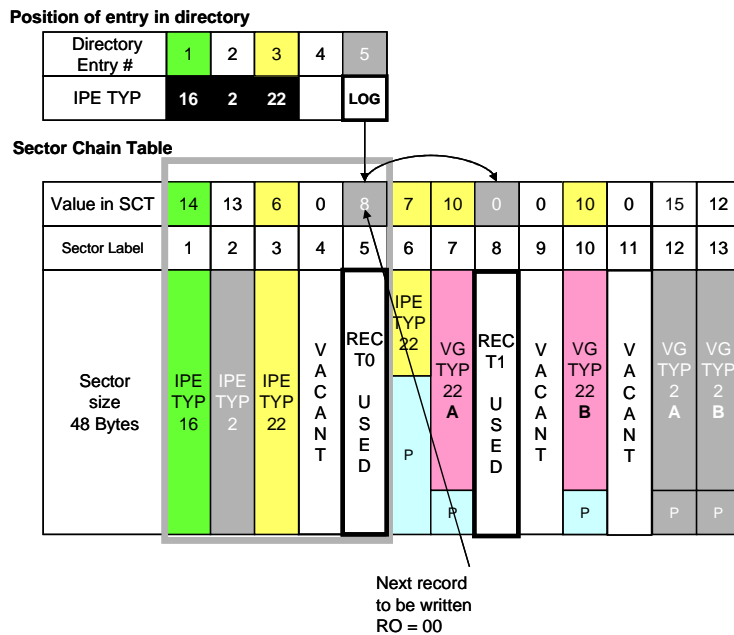


Figure 12c – Cyclic Log after writing the record T1

(Note the Sector Chain Table does not change)

The example in table 12c assumes that the use of the TYP16 IPE caused the Log entry to be written and Passback Time processing is determined by the CM.

Table 12c The directory entry associated with the overwriting of record T1 in the Log:

Byte offset #	Bit Position	Setting	Label	Description
0	7	bin1	LPF	Indicates Normal mode
0	6,5,4,3,2	bin00001	PTR	Pointer to the directory entry of the IPE most relevant to the last Log record created
0	1,0	bin00	EEI	Entry / Exit indicator
1	All	0xXXXXXX	DTS	Date Time Stamp
4	7,6	Bin00	RO	Record Offset
4	5,4,3,2,1,0	bin000111	PTLBM	Passback Time relating to the use of IPE pointed to by PTR

The value of R0 is set to bin 00 indicating that record T0 is the next record to be written. In this case Record T1 is the latest record.

5.1.6 Directory Sequence Number (DIRS#)

A single byte binary integer in the range 0 – 255 that shall be incremented to 1 more than the highest previous value of this Data Element every time the contents of the Directory Data Group are changed, when taking into account:

- the possible presence of another copy of the Directory Data Group (if software Anti-tear is used).
- roll over from FF(hex) to a value of 00 (hex) may occur.

Where there are two valid copies of the Directory Data Group that have sequence numbers of value FF(hex) and 00 (hex) respectively, then Rollover shall be deemed to have occurred and 00 (hex) shall be taken as the highest current value.

**5.2 Directory Instance Identifier (InstanceID)**

The Directory InstanceID structure shall hold Data Elements containing information that shall:

- Identify the key version to be used with the cryptography for the Directory Data Group’s Seal.
- Allow varying iterations of the fixed ISRN to be identified.
- Hold the unique identity of the last ISAM to generate the Directory Data Group’s SEAL.

The contents of the Directory InstanceID structure are shown in Table 8. Each Data Element is then described in turn.

**Table 8 - Coding of the Directory InstanceID**

Byte offset #	# of Bits	Data type	Label	Description of Data Element
0	4	HEX	KID	The identification of the key to use when verifying or generating the Directory Data Group Seal
0	4	HEX	INS#	Iteration number for the ITSO Shell Data Group
1	32	HEX	ISAMID	The unique ID of the ISAM that updated the Directory Data Group Seal

**5.2.1 Key Identifier (KID)**

A 4 bit binary integer in the range 0 – F (hex) shall be used to identify the version of the Key to be used when generating or verifying the Seal of the Directory Data Group. This integer is interpreted by the ISAM.

**5.2.2 ITSO Shell Iteration number (INS#)**

This 4 bit binary integer in the range 0 - 9 (hex) shall be used to indicate a new iteration of the same ITSO Shell. It is included in the Directory Data Group as, unlike the ITSO Shell Environment Data Group, it may be modified throughout the life of the ITSO Shell.

To reference a particular iteration of the same ITSO Shell the ITSO Shell reference string (ISRN & INS#) shall be used.

The ITSO Shell reference string shall be used in any form of ITSO Shell Hotlist when identifying a particular ITSO Shell.

Note: Incrementing the iteration number will ensure that previously Hotlisted ITSO Shells may be made valid again even though the previous iteration number may still be widely present on Hotlists. This method replaces the un-hot method specified in version 1 of this Specification.

In the event that the iteration number is incremented beyond 9 then Rollover to 0 shall occur. The values A - F (hex) are reserved for future use by ITSO.

**5.2.3 ITSO Security Application Module Identity (ISAMID)**

A 4 byte binary Data Element in the range 00 – FF,FF,FF,FF (hex) that uniquely identifies the Licensed Member and SAM that last changed this Directory Data Group.

Note: The ISAMID is made up of the OID of the Licensed Member to whom the ISAM is registered and a unique ISAM serial number. See Annex B for details.

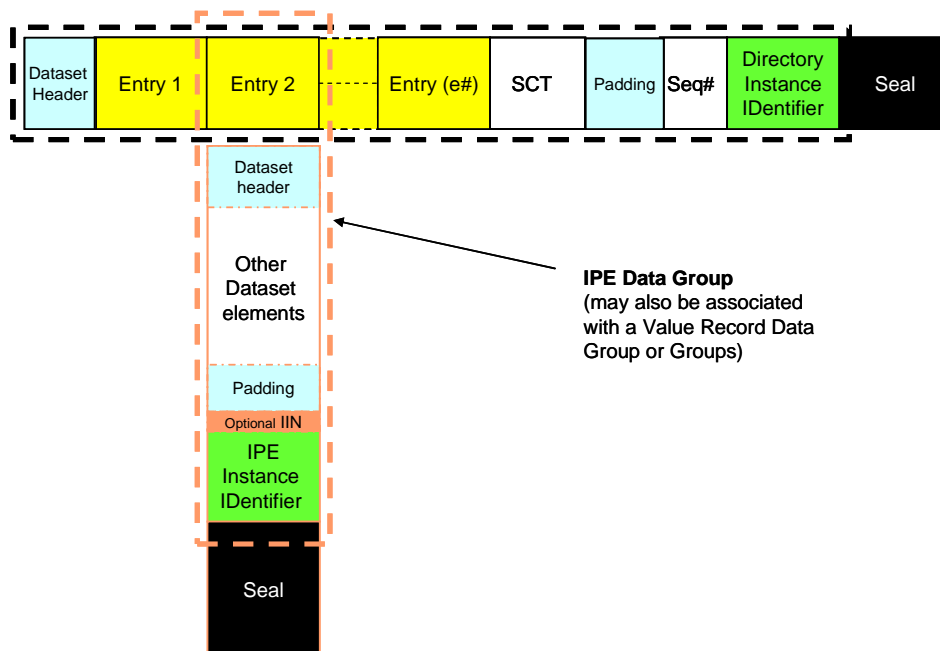
### 5.3 Directory Data Group Seal

The Directory Data Group Seal is a single variable length binary integer Data Element of at least 8 Bytes. It shall hold the result of a cryptographic process applied to the following data structures

- All other structures in the Directory Data Group.
- The ISRN and CM.

The data structures covered by the Directory Data Group Seal can be verified as valid by an ISAM in a POST. Also, if permitted, new Seals can be generated by the ISAM when a POST modifies a Directory Data Group. Cryptographic processes involved in the verification and generation of Seals are defined in ITSO TS 1000-7 and ITSO TS 1000-8.

All Directory Data Group Seals shall be generated and verified by an ISAM. The keys required shall be exclusively generated by the ITSO Security Management Service and, using secure class 3 messages, forwarded via an Asset Management Service (AMS) function (as defined in ITSO TS 1000-4) to every ISAM in the POSTs under the control of said AMS. The extent by which the Directory Data Group Seal binds to other Data Groups is illustrated by the bold broken lines in Figure 13.



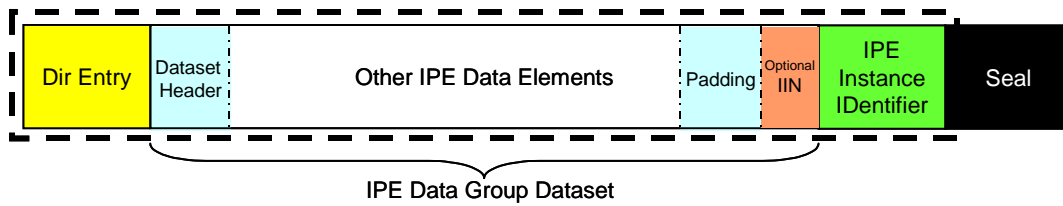
**Figure 13 - Relationship of the Directory Data Group Seal to other groups.**

The Seals for the IPE Data Group and the Directory Data Group both include the Label structure that forms the IPE Data Group’s Directory Entry. The overlap avoids duplicating elements in the IPE Dataset and gives added protection against Directory Entry misuse.

Note: Currently all the Seals defined by ITSO are 8 byte Integers. However ITSO does not preclude the use of longer Seals and different cryptographic algorithms in the future, whilst supporting compatibility with the current Seals for as long as required.

## 6. The IPE Data Group

The IPE Data Group consists of four structures: the Directory Entry; the IPE Dataset; the IPE InstanceID and the Seal. This is illustrated in Figure 14.



**Figure 14 - The IPE Data Group**

IPE Data Groups may be repeatedly added or removed during the life of an ITSO Shell. One or two copies of a Value Record Data Group may also be associated with an IPE Data Group.

### 6.1 IPE Directory Entry structure

This is the “Label” structure for the IPE Data Group and a single copy of this structure is shared with the Directory Data Group. It indicates: the owner of the IPE, the type and sub type of IPE and its expiry date. The entry is bound cryptographically to both the Directory and the IPE Data Groups.

The IPE Directory Entry structure may also be used to indicate the presence of a Private Application in which case:

- The OID Data Element shall indicate the owner of the Private Application.
- The TYP Data Elements shall be set to the value 0.
- The PTYP Data Element shall be as defined in the CMD.

All other Data Elements remain as defined herein.

#### 6.1.1 Coding of an IPE Directory Entry

The contents of the IPE Directory Entry are shown in Table 9. Each Data Element is then described in turn.

**Table 9 - Coding of an IPE Directory Entry**

Byte offset #	Bit Position	Data Type	Label	Description
0	7	FLAG	EF	OID Extension Flag
0	6 – 0	BIN	OID	Operator Identification MS seven bits
1	7 – 2	BIN	OID	Operator Identification LS six bits
1	1,0	TYP	TYP	IPE Type MS two bits
2	7 – 5	TYP	TYP	IPE Type LS three bits
2	4 – 0	PTYP	PTYP	IPE Sub Type
3	7	FLAG	VGP	Set to 1 if there is a Value Record Data Group associated with this Entry
3	6	FLAG	IINL	Set to 1 if the IIN for the Entry is not the ITSO Shell IIN
3	5 – 0	DATE	EXP	IPE Expiry date MS six bits
4	7 – 0	DATE	EXP	IPE Expiry date LS eight bits

**6.1.2 OID Extension Flag (EF)**

This Data Element shall contain a single flag bit that determines the range of OIDs supported (see Annex B)

**6.1.3 Operator Identification (OID)**

Subservient to the IIN, the value of the OID shall be supplied by the ITSO Registrar and in this context it shall:

- Be the identity number of the Operator that owns and is responsible for the IPE or Private Application associated with the Directory Entry.
- Normally be a 13 bit binary integer that may be extended, and if extended shall be interpreted as per Annex B of this Specification.

The values 0000, and from 1F40 to 1FFF (hex) shall be reserved for future use by ITSO.

**6.1.4 IPE Type (TYP)**

A 5 bit binary integer coded in the range 01 – 1F (hex), the value of which shall define the data content of an IPE.

Where the Directory Entry points to a Private Application then the value of TYP shall be set to 00 hex.

**6.1.5 IPE Sub Type (PTYP)**

A 5 bit binary integer coded in the range 00 – 1F (hex). The value of which is defined by the owner of the IPE to indicate the Business Rules for use of the IPE.

Where TYP equals 00 (hex) then this Data Element shall be defined separately for each CMD.

**6.1.6 Value Group Present flag (VGP)**

This Data Element contains a single flag bit which when set to 1 shall indicate the presence of a Value Record Data Group associated with the IPE. When this flag is set to 0 then no associated Value Record Data Group shall be present.

**6.1.7 IINL**

This Data Element contains a single flag bit which, when set to 0, shall indicate that the network to which the Operator that owns the IPE belongs is defined by the IIN of the ITSO Shell.

When set to 1, it indicates that the network to which the Operator that owns the IPE belongs shall be defined by the IIN in the IPE Dataset. Such an IINL setting requires that the optional IIN Data Element in the IPE Data Group shall be present.

**6.1.8 Expiry (EXP)**

A fourteen bit binary integer coded as a DATE data type. A value of 0 shall indicate a permanent IPE.

**6.2 IPE Dataset**

This data structure contains all Data Elements particular to the operation of the IPE and is subservient to the TYP Data Element held in the IPE Directory Entry structure.

All IPE Datasets use a common Dataset header template namely the first three Data Elements of the Dataset. The contents of an IPE Dataset are illustrated in Table 10.

**Table 10 - IPE Dataset Data Elements**

Byte offset #	# of bits	Data type	Label	Description of Data Element
0	6	BIN	IPELength	The IPE Dataset length (in Blocks)
0.75	6	BMP	IPEBitMap	The IPE BitMap
1.5	4	HEX	IPEFormatRevision	The Format Revision number of this IPE Dataset
AR	AR		IPE data	The remaining Data Elements in the IPE Dataset
AR	AR	PAD	PAD	Pad with zeros as required to whole Block boundaries inclusive of any succeeding IIN Data Element if present
AR	24	IIN	IIN	Optional IIN

**6.2.1 IPELength**

This Data Element is a 6 bit binary integer the value of which is the length (inclusive of this element and any padding elements) of the IPE Dataset to the start of first Data Element of the IPE InstanceID in Blocks, where a Block consists of BL bytes. BL is prescribed for each value of IPEFormatRevision as defined in ITSO TS 1000-5.

**6.2.2 IPEBitMap**

This is a set of six flag bits that indicate which Data Elements are present in the IPE. The IPEBitMap is defined separately in ITSO TS 1000-5 for each IPE Type.

**6.2.3 IPEFormatRevision**

This Data Element is a 4 bit binary integer, of value in the range 1 – 7 that indicates the revision number for this version of the IPE Data Group. This number is coded separately, in ITSO TS 1000-5, for each IPE Type.

**6.2.4.IPE Data Elements**

The remainder of the IPE Dataset is of variable length and shall be populated with Data Elements, some of which may be optional. These elements are defined separately in ITSO TS 1000-5 for each IPE Type.

**6.2.5 Padding**

This Data Element is of variable length that shall consist of as many binary zeros as are required to pad the IPE data set to whole Block boundaries inclusive of any succeeding Optional IIN if present.

**6.2.6 Optional IIN**

This Data Element is of 3 Bytes fixed length and is only present if the LSB of IPEBitmap is set to 1.

NB: It is appended after the padding because when present in this position it is interpreted by the ISAM as a logical extension of the Instance Identifier.

**6.3 IPE InstanceID**

The IPE InstanceID structure holds Data Elements containing information that shall:

- Identify the key version to use with the cryptography for the IPE Data Group Seal.
- Allow varying iterations of the IPE to be identified.
- Hold the unique identity of the ISAM or Proxy ISAM that created the IPE.
- Uniquely identify every IPE created.

The contents of the IPE InstanceID structure are shown in Table 11. Each Data Element is then described in turn.

**Table 11 - Coding of the IPE InstanceID**

Byte offset #	# of Bits	Data type	Label	Description of Data Element
0	4	HEX	KID	The identification of the key to use when verifying or generating the IPE Data Group Seal
0	4	HEX	INP#	Iteration number for the IPE Data Group
1	32	HEX	ISAMID	The unique ID of the ISAM that created the Seal of this IPE Data Group
5	24	HEX	ISAMS#	The Sequence Number that is incremented by one for each InstanceID created by the ISAM identified by the ISAMID in this structure

**6.3.1 Key Identifier (KID)**

A 4 bit binary integer in the range 0 – F (hex) that shall identify the version of the Key to be used when generating or verifying the Seal of the IPE Data group. This integer is interpreted by the ISAM.

**6.3.2 IPE Iteration number (INP#)**

This 4 bit binary integer in the range 0 - 9 (hex) shall be used to indicate a new iteration of the same IPE.

To reference a particular iteration of the same IPE the IPE reference string (ISAMID & ISAMS# & INP#) shall be used.

This IPE reference string shall be used in any form of IPE Hotlist that identifies a particular IPE.

In the event that the iteration number is incremented beyond 9 then Rollover to 0 shall occur. The values A - F (Hex) are reserved for future use by ITSO.

Note: Incrementing the iteration number will ensure that previously Hotlisted IPEs may be made valid again even though the previous iteration number may still be widely present on Hotlists

### 6.3.3 ISAM Identity (ISAMID)

A 4 byte binary Data Element in the range 00 – FF,FF,FF,FF (hex) which, in conjunction with the IIN, shall uniquely identify the Operator and ISAM or Proxy ISAM that created this IPE.

Note: The ISAMID is made up the OID of the Licensed Member to whom the ISAM is registered and a unique ISAM serial number. See Annex B for details. Proxy ISAMs are “virtual” ISAMs that may be installed inside a physical ISAM in order to create IPEs on behalf of “not on us” IPE Owners. Proxy ISAMs have limited functionality and cannot sign transactions; this can only be done by the physical ISAM within which the proxies are installed.

### 6.3.4 ISAM Sequence number (ISAMS#)

This is a 3 byte binary Data Element in the range 00 – FF,FF,FF (hex) which, in conjunction with the ISAMID, uniquely identifies the IPE. Its value shall be incremented by 1 every time an IPE is created and it shall never be reset. Once the value FF,FF,FF (hex) is reached the associated ISAM or Proxy ISAM is rendered inoperable.

## 6.4 IPE Data Group Seal

The IPE Data Group Seal is a single variable length binary integer Data Element of at least 8 bytes. It shall hold the result of a cryptographic process applied to the following Data Structures.

- All other structures in the associated IPE Data Group.
- The ISRN and CM.

The extent by which the Seal covers the IPE Data Group is illustrated by the broken line in Figure 14.

The data structures covered by the IPE Data Group Seal may be verified as valid by an ISAM in a POST. Also, if permitted, new Seals shall be generated by the ISAM when a POST creates or modifies an IPE Data Group. Cryptographic processes involved in the verification and generation of Seals are defined in ITSO1000 Parts 7 & 8. All IPE Data Group Seals shall be generated and verified by an ISAM. The keys required shall be exclusively generated by the ITSO Security Management Service and, using secure class 3 messages, forwarded via an AMS function (as defined in ITSO TS 1000-4) to ISAMs as required in the POSTs under the control of said AMS.

Note: Currently all the Seals defined by ITSO are 8 byte Integers. However ITSO does not preclude the use of longer Seals and different cryptographic algorithms in the future, whilst supporting compatibility with the current Seals for as long as required.

## 6.5 Orphan IPE Data Groups

Orphan IPE Data Groups are special cases of the IPE Data Group that exist in isolation from an ITSO Shell Environment or Directory Data Group they facilitate the efficient use of small memory capacity CM and provide a unique Seal for Transient Tickets. In these cases the Orphan IPE Data Group respectively acts as an:

- IPE;
- envelope for a Transient Ticket Record found in the Cyclic Log.

**6.5.1 The Orphan IPE Data Group as an IPE**

In this case the structure is identical to that defined in clauses 6.0 to 6.4 of this Specification with the following exceptions.

Where the CM does not support a directory the Label is held in isolation on the CM platform. The cryptography associated with the Seal binds the entire Orphan IPE Data Group to the CM.

The IPE Data Elements are as defined for the appropriate IPEs in ITSO TS 1000-5.

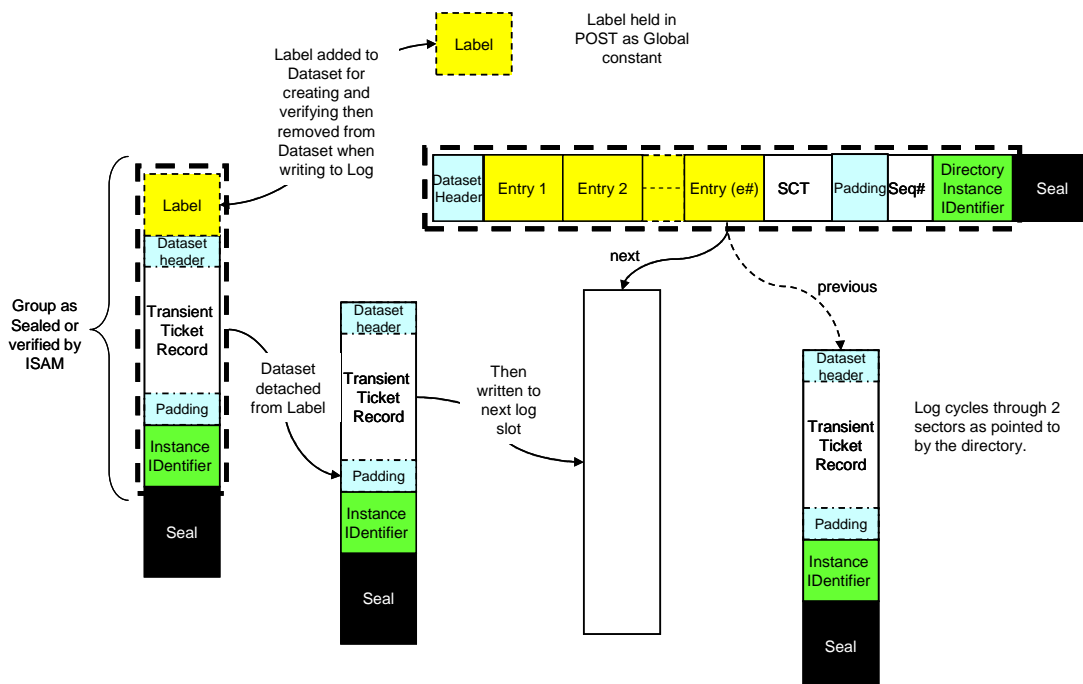
In all respects the Orphan IPE Data Group used as an IPE shall be interpreted as an IPE.

**6.5.2 The Orphan IPE Data Group as an envelope for a Transient Ticket Record**

In this case the structure is again identical to that defined in clauses 6.0 to 6.4 of this Specification with the following exceptions.

- the value of the Data Elements in the Label shall be fixed as defined in this clause.
- the Dataset shall contain the Transient Ticket Record Data as defined in ITSO TS 1000-5.
- the label is removed from the rest of the group prior to updating the Cyclic Log.

This process is illustrated in Figure 15



**Figure 15 Orphan IPE Data Groups and the Cyclic Log**

Rules for creating, verifying and storing the Transient Ticket Record in the Cyclic Log:

1. The POST shall use the fixed label in order to create and seal an Orphan IPE Data Group.
2. The POST shall then remove the Label from the Group prior to updating the Cyclic Log.
3. Updating the Log shall be performed in accordance with the rules defined in clause 2.4.8 of this Specification.
4. When verifying the authenticity of a Transient Ticket Record all the data from the Log shall be appended to the fixed Label supplied by the POST and then the Seal shall be verified.

### 6.5.2.1 Label contents

For this use of the Orphan IPE Data Group the Label contents shall be set to the values defined in Table 12

**Table 12 - Coding of the Label**

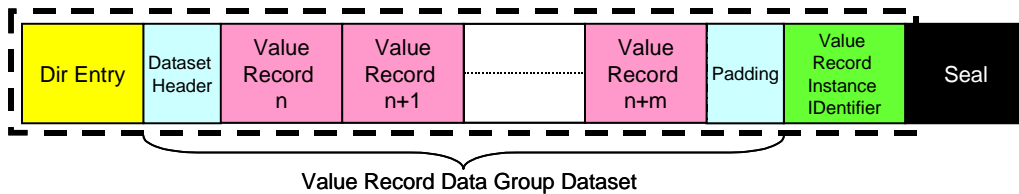
<b>Data Element</b>	<b>Value In hex</b>	<b>Description</b>
EF	0	OID Extension Flag
OID	1F, E0	Operator Identification
TYP	00	IPE Type
PTYP	00	IPE Sub Type
VGP	0	Value Group Present flag
IINL	0	
EXP	00,00	IPE Expiry date

Note: The value specified for OID in table 12 may, from time to time, be changed by ITSO but shall remain within the range 1F,E0 to 1F,E7

## 7 Value Record Data Group

Value Record Data Groups are typically associated with IPE Data Groups and share a common Label structure that is also the Directory Entry.

The Value Record Data Group consists of four structures: the Directory Entry; the Value Record Dataset; the Value record InstanceID and the Seal. The Value Record Dataset shall hold copies of previous Value Records of identical fixed length in a cyclic store. This is illustrated in Figure 16.



**Figure 16 - The Value Record Data Group**

Value Record Data Groups may be repeatedly added or removed during the life of an ITSO Shell. One or two<sup>7</sup> copies of a Value Record Data Group are typically associated with an IPE Data Group.

Where software Anti-tear is used then even and odd numbered Value Records shall be stored alternately in the A and B copies of the Value Record Data Group respectively.

### 7.1 Value Record Directory Entry

This is the Label structure for the Value Record Data Group and a single copy of this structure is shared with any associated IPE and the Directory Data Groups.

The Value Record Data Group shares the ownership and expiry date of the associated IPE Data group.

The Value Record Directory Entry shall be bound cryptographically to both the Directory Data Group and the IPE Data Group.

The elements that make up the Value Record group Directory Entry structure shall be identical to those specified in clause 6.1.

### 7.2 Value Record Dataset structure

The Value Record Dataset shall contain the current Value Record and one or more previous Value Records stored in a cyclic manner. The VGBitMap Data Element indicates how many of these records are supported.

The contents of the Value Record Dataset are shown in Table 13. Each Data Element is then described in turn.

<sup>7</sup> Where software Anti-tear is used

**Table 13 - Value Record Dataset Data Elements**

Byte offset #	# of bits	Data type	Label	Description of Data Element
0	6	BIN	VGLength	The Value Record Dataset length (in Blocks)
0.75	6	BMP	VGBitMap	The Value Record BitMap
1.5	4	HEX	VGFormat Revision	The Format Revision number of this Record Dataset
2	120	HEX	VR1	Value Record 1
15(n-1) +3	120	HEX	VRn	Value Record n
15n +3	AR	PAD	PAD	Padded with zeros as required to whole Block boundary

Note: All Value records (VR(i)) where i = 1 – n are defined in detail in ITSO TS 1000-5 and of a fixed length of 15 Bytes each.

**7.2.1 VGLength**

This Data Element is a 6 bit binary integer, the value of which shall be the length (inclusive of this element and any padding elements) of the Value Record Dataset to the start of first Data Element of the Value Record InstanceID in Blocks. Where a Block consists of BL bytes, BL is defined for each value of VGFormatRevision as defined in ITSO TS 1000-5.

**7.2.2 VGBitMap**

This is a set of six flag bits that indicate how many Value Records are supported in the Value Record Dataset.

The flag bits for specific conditions are coded as shown in Table 14, where X indicates that the setting of that flag is irrelevant to that particular condition.

**Table 14 - Value Record Bit Map**

Byte Offset	0		1			
	1	0	7	6	5	4
Condition	MSB					LSB
RFU	0	0	0	0	0	0
1 Value Record supported	1	0	0	0	0	0
2 Value Records supported	1	1	0	0	0	0
3 Value Records supported	1	1	1	0	0	0
4 Value Records supported	1	1	1	1	0	0
5 Value Records supported	1	1	1	1	1	0
RFU	X	X	X	X	X	1

Only the Most Significant 5 Bits of the VGBitMap are currently defined. The remaining flag shall be set to zero and is RFU by ITSO.

**7.2.3 VGFormatRevision**

This Data Element is a 4 bit binary integer in the range 9 – 15 that shall indicate the revision number for this version of the Value Record Data Group. This number is defined separately in ITSO TS 1000-5 for each Value Record Data Group.

**7.2.4. Value Record Data Elements (VR(i))**

Value Records shall be of identical fixed length and a number of them (VR1 to VRn) populate this area. There are a number of common Data Elements in all forms of Value Record and these are shown here. Value record Data Elements associated with each IPE Type are defined in ITSO TS 1000-5. The content of each Value Record is shown in Table 15 with the common elements highlighted. Each Data Element is then described in turn.

**Table 15 - Value Record Data Elements**

Byte offset #	# of bits	Data type	Label	Description of Data Element
0	4	HEX	EventTypeCode	Type of Transaction (Load, spend, Auto load ....)
0 – 1	12	HEX	TS#	Transaction Sequence Number, incremented by 1 every time a new Value Record is recorded
2 – 4	24	DTS	DTS	The date and time the Transaction took place
5 – 8	32	HEX	ISAMID	The ISAMID of the POST that changed this record
9	8	HEX	ActionSequence Number	Sequence number for Actionlists
10	40	HEX		Value Record Data Elements defined in ITSO TS 1000-5.

**7.2.4.1 Type of Transaction**

As defined in ITSO TS 1000-5.

**7.2.4.2 Transaction Sequence Number (TS#)**

A 12 bit binary integer in the range 0 – F,FF (hex) that shall be incremented by 1 every time a new Value Record is recorded.

In the event that this value rolls over from F,FF (hex) it shall be set to 0,00 (hex).

An initial value of TS# = 0,00 (hex) shall apply to all Value Records present when creating one or more copies of the Value Record Data Group.

**7.2.4.3 Date Time Stamp (DTS)**

The DTS Data Element is coded as data type DTS and shall be loaded with the value of the DTS in the POST at the time when the Value Record was recorded.

**7.2.4.4 ISAM Identity (ISAMID)**

A 4 byte binary Data Element in the range 0 – FF,FF,FF,FF (hex) that shall uniquely identify the Operator and ISAM that changed this Value record.

Note: The ISAMID is made up the OID of the Licensed Member to whom the ISAM is registered and a unique ISAM serial number. See Annex B for details.

### 7.2.4.5 Action Sequence Number (ActionSequenceNumber)

As defined in ITSO TS 1000-3 clause 6.4.2.5.1 and qualified in ITSO TS1000-4 clause 7.5.3.2.

## 7.3 Value Record InstanceID structure

The Value Record InstanceID structure holds Data Elements containing information that shall:

- identify the key version to use with the cryptography for the Value Record Data Group Seal.
- hold the unique identity of the ISAM or Proxy ISAM that created the Value Record Data Group.
- uniquely identify every Value Record Data Group created.

### 7.3.1 Coding of the Value Record InstanceID

The content of the Value Record InstanceID structure is identical to that defined in clause 6.3 for the IPE InstanceID structure with one exception: Where the Value Record Data Group is created at the same time as the associated IPE Data Group, the Value Record Data Group shall be created first and where required by the Anti-tear mechanism duplicated on the CM. In this case the sequence number (ISAMS#) shall be 1 less than that contained in the associated IPE.

Note 1: The INP# Data Element in the Value Record InstanceID shall remain identical to the INP# element in the associated IPE Data Group. This means that if the INP# of the IPE Data Group is modified then the INP# within the associated Value Record Data Groups shall also be modified.

Note 2: If other Data Elements in the associated IPE are modified then the InstanceID in the Value Record Data Group shall also be modified to reflect any changes made to the IPE Data groups InstanceID (taking in to account the ISAMS# numbering requirements of this clause).

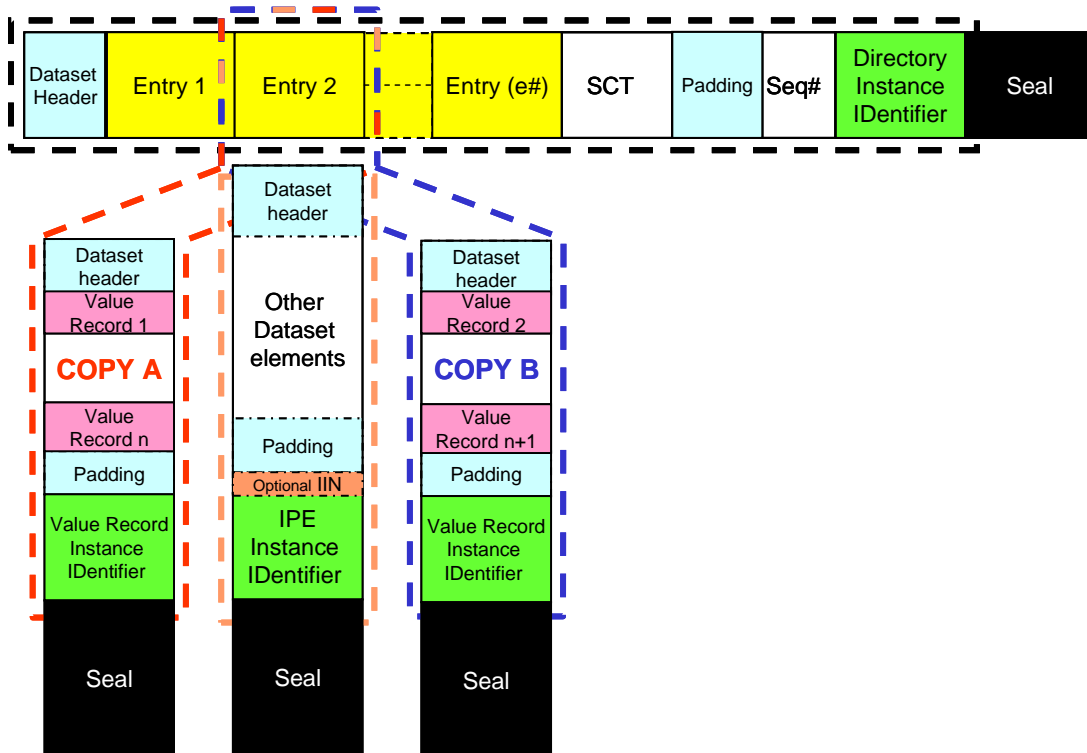
Note 3: Where there is a second copy of a Value Record Group. The Instance Identifier shall be identical in both Value Record Data Groups.

## 7.4 Value Record Seal

The Value Record Seal structure forms a cryptographic bond between the following:

- All other structures in the Value Record Data group.
- The associated IPE Data Group.
- The ISRN and CM.

The data structures covered by the Value Record Seal may be verified for validity by an ISAM in a POST. Also, if permitted, new Seals shall be generated by the ISAM when a POST creates or modifies a Value Record Data Group. The extent to which the Value Record Seal covers the Data Group and its relationship with its associated IPE Data group and Anti-tear copy (if present) is illustrated by the broken lines in Figure 17.



**Figure 17 - Relationship between the Value Record Data Group and other Data Groups**

The Value Record Data Group Seal is a single variable length binary integer of at least 8 bytes. It shall hold the result of a cryptographic process applied to all the data structures covered by the Seal.

All Value Record Data Group Seals shall be generated and verified by an ISAM. The keys required shall be exclusively generated by the ITSO Security Management Service and, using secure class 3 messages, forwarded via an AMS function (as defined in ITSO TS 1000-4) to ISAMs as required in the POSTs under the control of said AMS.

Cryptographic processes involved in the verification and generation of Seals are defined in ITSO1000 Parts 7 & 8...

Note: Currently all the Seals defined by ITSO are 8 byte Integers. However ITSO does not preclude the use of longer Seals and different cryptographic algorithms in the future, whilst supporting compatibility with the current Seals for as long as required.

## 8 Log Directory Entries

The Cyclic Log may be used to hold tickets and other events. It holds records that may be created when an IPE Data Group is used and has a Directory Entry known as the Log Directory Entry. In order to support a variety of logging options the Log Directory Entry has two modes of operation. Normal mode, which indirectly references the Log record created with the last transaction, and Basic Mode, where only the Log Directory Entry is updated and no Log record is created.

### 8.1 The Log Directory Entry

The Log Directory Entry has two constructions namely Normal and Basic.

- In normal mode the IPE referred to by the PTR Data Element is the IPE associated with the latest Log record.
- In basic mode the IPE referred to by the PTR Data Element is not linked to any Log record but was the IPE that caused the basic mode directory entry to be written

#### 8.1.1 Coding of the Log Directory Entry

The contents of the Log Directory Entry are shown in Table 16. Each Data Element is then described in turn.

**Table 16 - Coding of a Log Directory Entry**

Byte offset #	Bit Position	Data Type	Label	Description
0	7	FLAG	LPF	Indicates Normal or Basic mode
0	6,5,4,3,2	BIN	PTR	Pointer to an IPE
0	1,0	BIN	E EI	Entry / Exit indicator
1	All	DTS	DTS	Date time stamp
4	7,6	BIN	RO	Record Offset
4	5,4,3,2,1,0	BIN	PTLBM	Passback Time

#### 8.1.2 Log pointer flag (LPF)

This Data Element shall contain a single flag bit that shall indicate Normal or Basic mode of this Log Directory Entry.

For Basic mode: the flag is set to 0;

For Normal mode: the flag is set to 1.

#### 8.1.3 Pointer (PTR)

This Data Element contains a 5 bit binary integer in the range 00 – 1F (hex)

For Both modes: It shall contain the Directory Entry array reference number E(i) of the most relevant IPE used during the transaction which shall be the same as the IPE Pointer (where present) in the latest Log Record.



## Annex A (informative) CRC Generation

### A.1 Example of CRC generation

This Annex is provided for explanatory purposes and indicates the bit patterns that will exist in the physical layer. It is included for the purpose of checking an implementation of CRC encoding. Refer to ISO/IEC 13239 and CCITT X.25 #2.2.7 and V.42 #8.1.1.6.1 for further details.

Initial Value = 'FFFF'

#### A.1.1 Examples of bit patterns

##### A.1.1.1 Example 1

Transmission of first byte = '00', second byte = '00', third byte = '00', CRC\_B appended.

Calculated CRC\_B = 'C6CC'

1st byte	2 <sup>nd</sup> byte	3rd byte	CRC_B	
'00'	'00'	'00'	'CC'	'C6'

**Figure A1 - Example 1 for CRC\_B encoding**

##### A.1.1.2 Example 2

Transmission of first byte = '0F', second byte = 'AA', third byte = 'FF', CRC\_B appended.

Calculated CRC\_B = 'D1FC'

1st byte	2 <sup>nd</sup> byte	3rd byte	CRC_B	
'0F'	'AA'	'FF'	'FC'	'D1'

**Figure A2 - Example 2 for CRC\_B encoding**

##### A.1.1.3 Example 3

Transmission of first byte = '0A', second byte = '12', third byte = '34', fourth byte = '56', CRC\_B appended.

Calculated CRC\_B = 'F62C'

1st byte	2nd byte	3rd byte	4th byte	CRC_B	
'0A'	'12'	'34'	'56'	'2C'	'F6'

**Figure A3 - Example 3 for CRC\_B encoding**

## A.1.2 Code sample written in C language for CRC calculation

This example is reproduced from ISO/IEC 14443-3 and includes two CRC calculations A and B. All CRC's defined in this Specification are of the CRC\_B variety.

Note: Any users of type A cards will also be required to calculate the CRC shown here as CRC\_A.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define CRC_A 1
#define CRC_B 2
#define BYTE unsigned char

unsigned short UpdateCrc(unsigned char ch, unsigned short *lpwCrc)
{
    ch = (ch^(unsigned char)((*lpwCrc) & 0x00FF));
    ch = (ch^(ch<<4));

    *lpwCrc = (*lpwCrc >> 8)^((unsigned short)ch << 8)^((unsigned
short)ch<<3)^((unsigned short)ch>>4);

    return(*lpwCrc);
}

void ComputeCrc(int CRCType, char *Data, int Length,
    BYTE *TransmitFirst, BYTE *TransmitSecond)
{
    unsigned char chBlock;
    unsigned short wCrc;

    switch(CRCType) {
        case CRC_A:
            wCrc = 0x6363; /* ITU-V.41 */
            break;
        case CRC_B:
            wCrc = 0xFFFF; /* ISO/IEC 13239 (formerly ISO/IEC 3309) */
            break;
        default:
            return;
    }

    do {
        chBlock = *Data++;
        UpdateCrc(chBlock, &wCrc);
    } while (--Length);
    if (CRCType == CRC_B)
        wCrc = ~wCrc; /* ISO/IEC 13239 (formerly ISO/IEC 3309) */

    *TransmitFirst = (BYTE) (wCrc & 0xFF);
    *TransmitSecond = (BYTE) ((wCrc >> 8) & 0xFF);

    return;
}

BYTE BuffCRC_A[10] = {0x12, 0x34};
BYTE BuffCRC_B[10] = {0x0A, 0x12, 0x34, 0x56};
unsigned short Crc;
BYTE First, Second;
FILE *OutFd;
int i;

int main(void)
{
    printf("CRC-16 reference results 3-Jun-1999\n");
```

```
printf("Crc-16 G(x) = x^16 + x^12 + x^5 + 1\n\n");

printf("CRC_A of [ ");
for(i=0; i<2; i++) printf("%02X ",BuffCRC_A[i]);
ComputeCrc(CRC_A, BuffCRC_A, 2, &First, &Second);
printf("] Transmitted: %02X then %02X.\n", First, Second);

printf("CRC_B of [ ");
for(i=0; i<4; i++) printf("%02X ",BuffCRC_B[i]);
ComputeCrc(CRC_B, BuffCRC_B, 4, &First, &Second);
printf("] Transmitted: %02X then %02X.\n", First, Second);

return(0);
}
```

## Annex B (normative) OID numbering

This Annex is included to illustrate the coding of the OID reference number in the various contexts in which it is used.

### B.1 Roles of organisations

An OID shall be supplied to a Licensed Member by the ITSO Registrar and is used throughout the ITSO Environment to identify those Licensed Members responsible for the major ITSO roles.

**Table B1 - Major ITSO roles**

Role #	Role	ITSO Entity
1	Provision of the ITSO Shell (or CM and ITSO Shell)	ITSO Shell Owner
2	Provision of the ITSO IPEs	IPE Owner
3	Accepting the IPEs	Service Operator
4	Distributing the IPEs and ITSO Shells	IPE / ITSO Shell Retailer

It is envisaged that some Licensed Members will fulfil all four roles whereas others will own, accept and distribute IPEs whilst some will only act as Service Operators and IPE / ITSO Shell Retailers. It is anticipated that there will be more organisations accepting and distributing IPEs than those owning IPEs or ITSO Shells.

Note: Any Licensed Member may also use their own OID when performing a Collection and Forwarding role. The roles described here are as defined in the emerging CEN TC 278 standard entitled Interoperable public transport fare management system architecture working draft 2003-06-03. Where a Licensed Member carries out the Collecting and Forwarding role only, they may register for an OID solely applicable to roles 3 and 4.

#### B 1.1 Initial OID range

Because of the limitations of the ISO numbering scheme when applied to ITSO Shells and the need to limit the code space occupied by Directory Entries, version 1 of this Specification limited the entire OID number range to 0001- 8000<sup>8</sup> independent of the role of the organisation.

This range is now deemed restrictive and in the light of other changes made to the Specification an additional bit has now be allocated to flag extended significance to the range of OIDs.

---

<sup>8</sup> There are some numbers in the range 8001 - 8191 that have special significance or are reserved for future use by ITSO

### B 1.2 New ranges supported

By using the Extension Flag bit (EF) to indicate an extension of the OID in conjunction with the remainder of the ISAM ID Data Element further OID numbering ranges are made available. These numbers are however context sensitive as follows

The number of OIDs that can be allocated to ITSO Shell Owners cannot be increased and is still restricted to 8000. OIDs within this range may also be used to identify IPEs Owners, Service Operators and IPE and ITSO Shell Retailers...

The range of OIDs used to identify IPE Owners can be extended by a further 8192 numbers. These additional OIDs cannot be used to identify ITSO Shell Owners but may also be used to identify Service Operators and IPE and ITSO Shell Retailers.

A range of OIDs that can only be used to identify Service Operators and ITSO Shell and IPE Retailers has been introduced providing an additional 16384 numbers for this purpose. These additional OIDs cannot be used for identifying ITSO Shell or IPE Owners.

**Table B2 - OID numbering ranges**

<b>Roles allowed</b>	<b>OID Numbering range</b>	<b>Description</b>	<b># of ISAMs per OID</b>
1,2,3,4	0001 to 8000	Can be a CM / ITSO Shell Owner, IPE Owner, Service Operator and ITSO Shell and IPE Retailer	262,142
2,3,4	8192 to 16383	Can be a IPE Owner, Service Operator and ITSO Shell and IPE Retailer	131,070
3,4	24576 to 32767	Can be a Service Operator and ITSO Shell and IPE Retailer only	65534
3,4	57344 to 65535	Can be a Service Operator and ITSO Shell and IPE Retailer only	65534

Note: The shaded area in the Table B2 indicates the additional numbers available for allocation and the restrictions that apply.

The numbering system is contiguous and inclusive within the ranges shown in the table. However the gaps between the ranges shall not be used as these ensure correct decoding of the extension flag and backwards compatibility with the earlier system.

### B 1.3 Mechanisation of the extended numbering system

The OID is constructed of 13, 14 or 16 bits incorporated within the ISAM ID. The ISAM ID is a four byte Data Element and its significance is interpreted in accordance with the following rules.

When the EF bit is set to 0 the 14<sup>th</sup> bit of the ISAM ID is also set to 0 and the OID shall have 13 significant bits and the remaining 18 bits are the rest of the ISAM serial number

When the EF bit is set to 1 the 14<sup>th</sup> bit of the ISAM ID is also set to 1 and the OID shall have 14 or 16 significant bits as determined by the state of the 15<sup>th</sup> bit of the ISAM ID.

- and If bits 15 and 16 respectively are set 0,x then the OID has fourteen bits of significance and the remaining 17 bits are the rest of the ISAM serial number (x means either state is valid).
- and If bits 15 and 16 respectively are set 1 0 then the OID has sixteen bits of significance and the remaining 16 bits are the rest of the ISAM serial number.
- and If bits 15 and 16 respectively are set 1 1 then the OID has sixteen bits of significance and the remaining 16 bits are the rest of the ISAM serial number.

This system means that the number of ISAMs available for a given OID is reduced by a half, one quarter or one eighth respectively compared to version 1 of the Specification.

