Contactless Fare Media System Standard

Part II - Contactless Fare Media Data Format and Interface Standard

(APTA IT-UTFS-S-002-06)

Version 1.0
October 8, 2006

Note: Document renumbered March 2013, previously referenced as APTA S-UTFS-WP1-001-06. No content was changed.

Prepared by members of the Work Package - 1 group of the Financial Management Committee of the American Public Transportation Association (APTA) Universal Transit Fare System (UTFS) Task Force

The APTA Rail Standards Policy and Planning Committee approved this standard for release on October 8, 2006.

American Public Transportation Association
1666 K Street, NW
Washington, DC 20006-1215, USA

Abstract: This standard provides the basic steps and considerations that should be employed in order to define, implement, and manage a security program for a regional smart card-based fare collection system.

Keywords: fare collection, transit, public transportation, smart card

Introduction

(This Introduction is not part of the APTA IT-UTFS-S-002-06 Standard)

This Standard is part II (Part II) of a suite of standards/guidelines that together form the Contactless Fare Media System Standard (Standard). This and other parts of the Standard include the following:

— Part I-Introduction and Overview (Part I)

— Part II-Contactless Fare Media Data Format and Interface Standard (Part II)

— Part III-Regional Central System Interface Standard (Part III)

— Part IV-Automated Fare Collection System Security Planning and Implementation Guidelines and Best Practices (Part IV)

— Part V-Compliance Certification and Testing Standard (Part V)

The parts of the Standard noted above are intended to be implemented as a package to complete an end-to-end integration of fare collection information processing to best provide an interoperable system. Detailed descriptions of all the parts of the Standard can be found in Part I - Introduction as well as within the introduction sections of each part.

The application of any standards, practices or guidelines contained herein is voluntary. In some cases, federal and/or state regulations govern portions of a rail transit system's operation. In those cases, the government regulations take precedence over this Standard/Guideline. APTA recognizes that for certain applications, the Standards or Practices or Guidelines, as implemented by rail transit systems, may be either more or less restrictive than those given in this document.

The intent of this Part II of the Standard is to provide a consistent and uniform method for storing, retrieving and updating data from contactless fare media used in transit applications. By applying the Standard/Guideline to the design of a new fare collection system or upgrade of an existing system, combined with adherence to a set of regional implementation needs or security and operating rules, interoperability with other compliant systems may be achieved.

Document Development Process

Development of this Standard and its parts was guided by the APTA Universal Transit Fare System (UTFS) Task Force and its bylaws. It is the mission of the Task Force to develop a series of documents that provides industry guidance for the creation of an open architecture payment environment that promotes greater access and convenience to the public transportation network and enables integration of independent payment systems. To accomplish this mission, the Task Force membership established a broad representation of the transit industry specifically including transit system operators, the Federal Transit Administration (FTA), manufacturers, engineering and consulting firms, transit labor organizations and others with an interest in the revenue management aspects of the transit industry.

To be effective and responsive to transit industry needs, the Task Force in its effort to develop fare collection standards relies on the following guiding principles:

— Promote economies of scale for agencies and enable more competitive procurements,

— Provide a platform to support agency independence and vendor neutrality,

— Strive for an open architecture environment for hardware and software utilizing commercially available products,

— Foster development for a multi-modal and multi-application environment, and

— Provide guidelines to assist in making informed decisions in the development of partnership strategies.

Applying these guidelines and relying on a broadly consensus driven decision process has produced these important industry-based standards.

To be successful, any consensus process involving organizations with diverse interests must have rules defining the procedures to be used. APTA developed a set of bylaws the APTA UTFS Bylaws (Bylaws) as revised September 1, 2005 to govern the process. These bylaws contain the following basic principles:

— Membership open and broadly representative of industry

— Open process and open meetings

— Consensus based (defined as 75% super-majority)

— Mandatory minimum public comment period

— Response required to all reasonable comments received

— Final approval voting based on one vote per organization

— Maximum use of electronic communication

— The policy committee retains implementation authority

The bylaws and resulting process APTA used to develop these standards followed the process required by the American National Standards Institute (ANSI) to obtain ANSI Standards Development Organization (SDO) certification.

The specific approach of the Task Force for standard development is based on a consensus driven process broadly representing all the major revenue management industry groups and stakeholders. Figure (i) is an organizational diagram depicting the relationships that have been established to develop, to approve and to implement revenue management standards, recommended practices and guidelines.

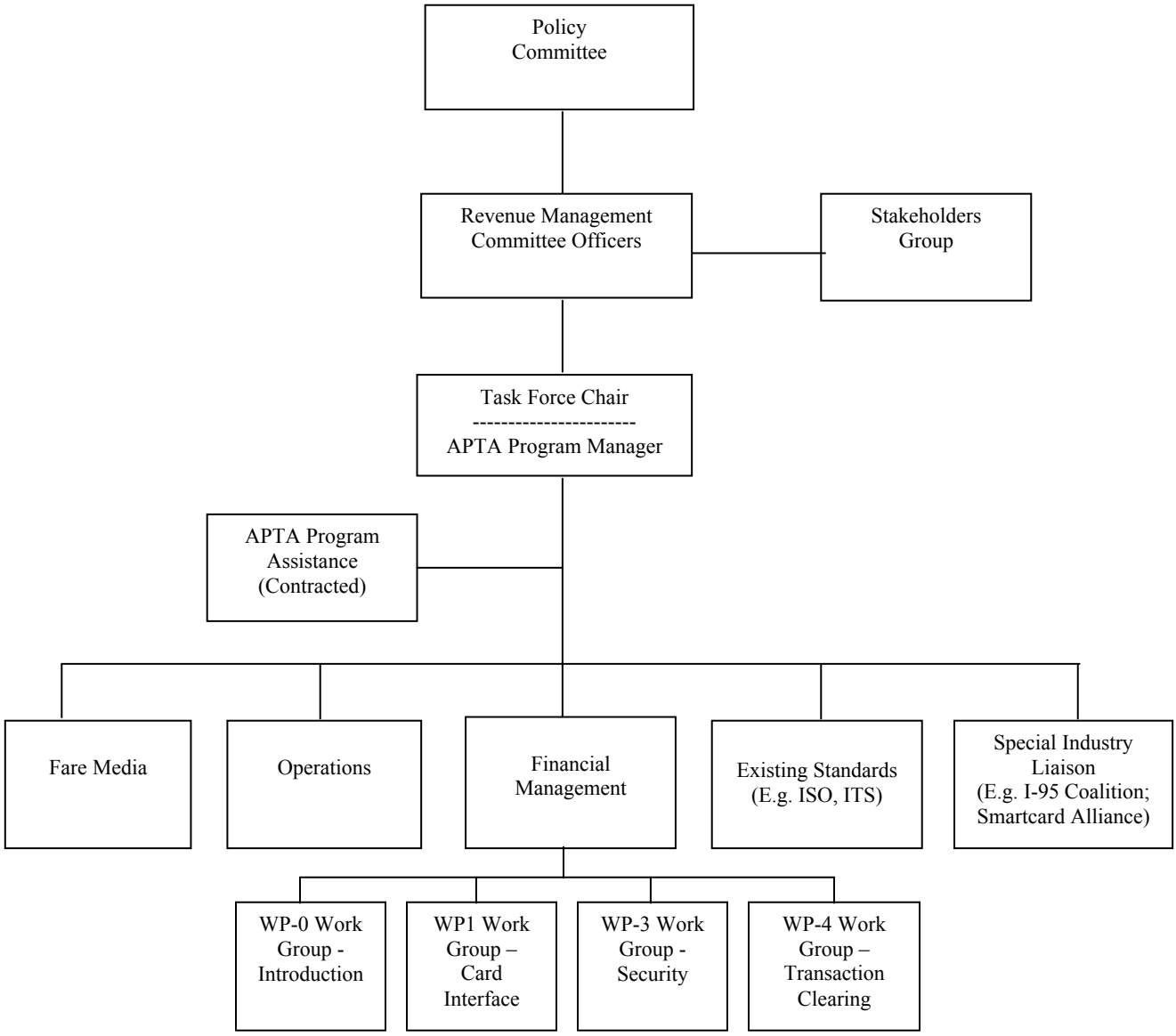


Figure (*i*) Universal Transit Fare System Standards Organization

The broad policies followed by the Task Force are set by the Rail Standards Policy and Planning Committee (Policy Committee) with oversight by the APTA Standards Development and Oversight Council (SDOC). APTA ensures that the policies set by the Policy Committee are followed. The officers of the Revenue Management Committee assist APTA staff in the implementation of policies set by the Policy Committee. The Task Force is organized into committees based on the priorities set by the stakeholders group and Revenue Management Committee officers and approved by the Policy Committee. Task Force committees develop individual work plans and schedules. Task Force committees may divide into sub-committees or working groups of subject matter experts to develop initial drafts of individual standards or recommended practices.

Given the consensus driven decision process of the Task Force, voting and balloting on release of this document for consideration by the APTA Rail Standards Policy and Planning Committee was approved using the following conditions:

— A quorum of at least sixty percent (60%) of the total Task Force voting members participated for a valid vote to take place.

— Approval of this document required a super majority of 75% of the voting members that cast ballots (do not abstain) to vote in the affirmative for the Task Force to approve this document for release.

The document approval process necessary for release of an APTA standard follows the flowchart depicted in Figure (ii) as documented in the APTA UTFS Bylaws (Bylaws) as revised September 1, 2005 maintained and controlled by APTA. The Bylaws also provide policies on Task Force and committee organizational structure and document balloting requirements as noted above.

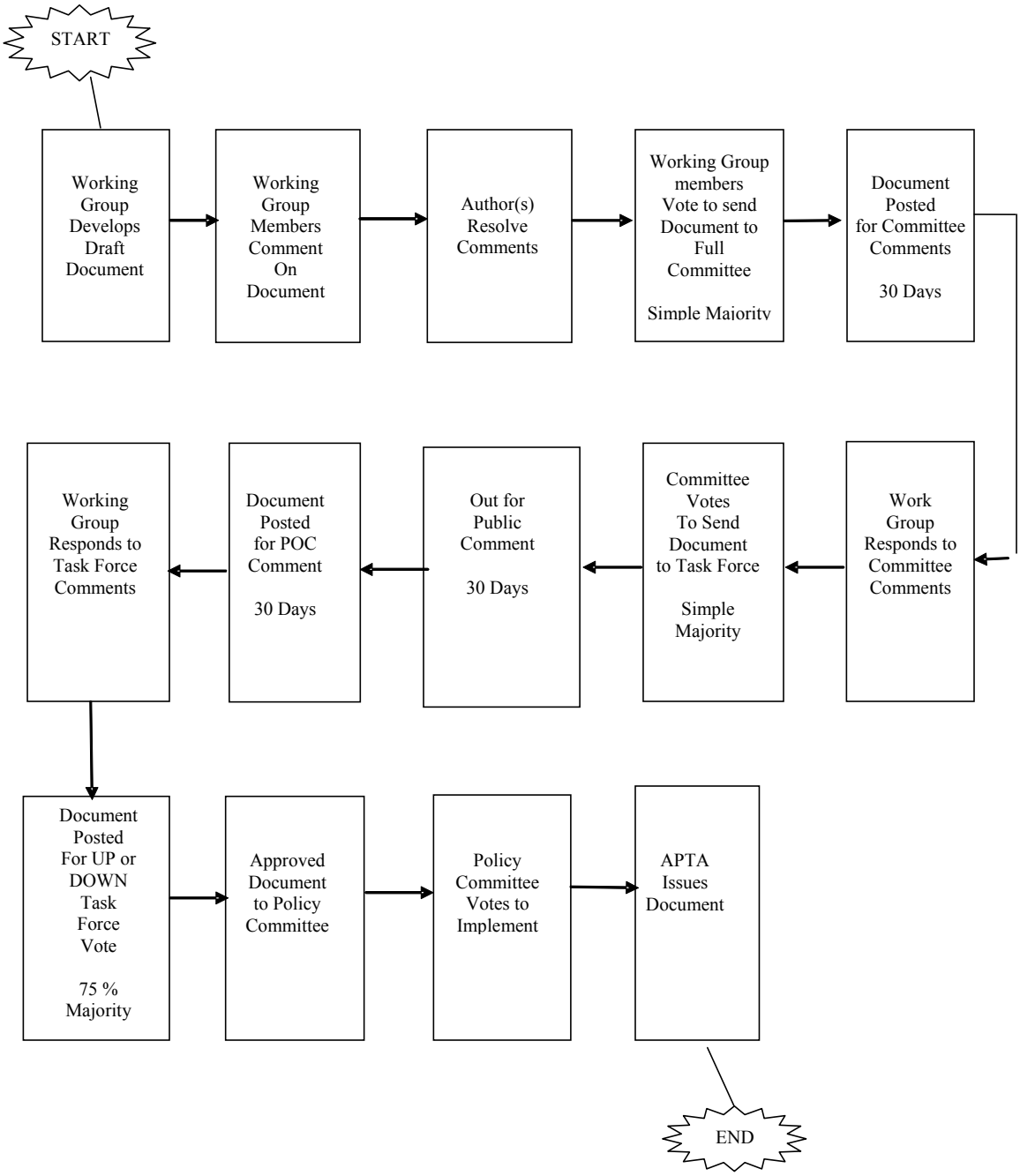


Figure (*ii*) Document Comment and Approval Process

Intellectual Property Provisions

To protect those offering technology during development of the Standard or Guideline and to protect those using the Standard or Guideline from copyright and patent infringements, the UTFS Task Force has implemented an Intellectual Property Policy. The inclusion of intellectual property provisions addressing patents, copyrights or trademarks is in accordance with APTAs Universal Transit Farecard Standards Intellectual Property Policy and Procedures, issued September 1, 2005, and enforced beginning October 17, 2005. The terms of this IP Policy are subject to the Universal Transit Farecard Standards Task Force Bylaws and in accordance with APTA Scope document, "APTA Universal Transit Farecard Standard Work Scope Specification, ATPA UTFS-D-TC-01A-05." All other documents, besides the Bylaws, concerning UTFS IP policies and procedures are controlled by this IP Policy, and other documents shall have no effect on the interpretation of the IP Policy.

Under this policy all participants in the APTA UTFS program including but not limited to transit agencies, fare collection system suppliers, financial institutions, consultants and other third party application providers shall submit a Letter of Acknowledgement, which states that, on behalf of the Organization with which they are affiliated and/or themselves, they have received and reviewed the IP Policy, and acknowledge that their participation in the UTFS standards development process, and the standard(s) adopted in the course of this process, will be subject to the IP Policy. Under this policy contributors are required to make known any patents, copyright material or other intellectual property that may be contained within the standard or essential to the standard. If contributors have intellectual property such as patents or copyright material contained within the standard/guideline, the IP Policy requires submission of a Letter of Assurance stating the terms and conditions for use of such intellectual property.

APTA further issues a call-for-patents during its public comment period prior to release of the Standard or Guideline.

Further, federal antitrust laws prohibit contracts, combinations and conspiracies in restraint of trade. Sanctions for violating the antitrust laws include civil damages (including treble damages) and criminal fines and imprisonment. The Policy of the American Public Transportation Association and the Task Force is to strictly adhere to the antitrust laws.

Standards vs. Guidelines/Recommended Practices

APTA develops standards and recommended practices/guidelines, and such distinction between these document types needs to be clear.

Characteristics of a Standard

A standard should be developed when the document:

a) Covers a system, component, process or task that is safety critical, or

b) Ensures interoperability between parts or equipment, or

c) Standardizes a design or process, or

d) Addresses an FRA or NISB concern, or

e) May become part of a regulation.

Characteristics of a Guideline/Recommended Practice

A recommended practice/guideline should be developed when:

a) The document describes only one of several acceptable approaches, or

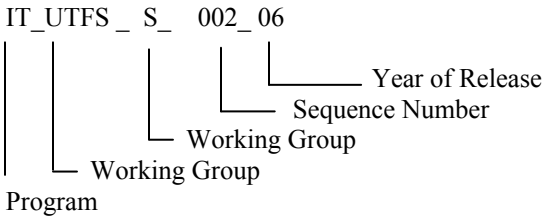
b) The document is tutorial in nature, or

c) The document does not meet one of the characteristics for a standard, or

d) Consensus could not be reached that the document should be a standard.

Document Numbering Nomenclature

Document numbering is composed of five parts. The first part designates the standard program the document falls under, in this case IT or Information Technology. The second part designates the working group or application where the standard was developed; which for this Standard is UTFS. The third part designates the type of document. A prefix "S" represents a general standard while recommended practices carry the prefix "RP" and Guidelines carry the prefix "GL." Finally, the last two sections attribute a document sequence number and the year the document was first released, respectively.



Document Maintenance & Requests for Revisions

APTA will review and update this document on an as needed basis, but at a minimum will review once every two years. The UTFS Task Force has responsibility for conducting reviews, addressing requests or suggestions for document revision or expansion and for implementing changes or revisions.

Requests for revisions of APTA standards and recommended practices/guidelines are welcomed from any interested party. Suggestions for changes to documents should be submitted in the form of a proposed change to the text along with the appropriate supporting documentation / rationale for the change.

Occasionally, questions may arise concerning the meaning of portions of these standards/guidelines as they are specifically applied. APTA will clarify such issues as necessary through the UTFS Task Force and the Rail Standards Policy and Planning Committee. Address comments, questions on interpretation or requests for changes to:

UTFS Staff Advisor
American Public Transportation Association
1666 K St., NW, 11th Floor
Washington, DC 20006

To obtain copies of this standard contact:

Information Center
American Public Transportation Association
1666 K St., NW, 11th Floor
Washington, DC 20006

Patents

Attention is called to the possibility that implementation of this guideline may require use of subject matter covered by patent rights. By publication of this guideline, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The APTA shall not be responsible for identifying patents or patent applications for which a license may be required to implement an APTA standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Participants

The American Public Transportation Association (APTA) greatly appreciates the contributions of Thomas Parker, Chair of UTFS Task Force and the following individuals who provided the primary effort in the drafting of this Standard.

Will Barley
Richard Barrett, chair FMC
Walt Bonneau, past WP1 chair
Lisa Bucci
Reid Holmes
Barney Louie
Brian Monk
David McIlwraith
Robert Murray
Gerard Najman
Tomas Oliva
Alex Pi, past WP1 chair
Brian Stein
Gary Yamamura

At the time this Standard was completed, the Work Package 1 Group responsible for the major development of this Standard included the following membership:

Tomas Oliva, C*hair*
Work Package – 1

Will Barley
Walt Bonneau
Lisa Bucci
Cynthia Chin Paik
Amber Dalzen
Kurt Elste
Bob Hamilton
Reid Holmes
Kevin Krest
Barney Louie
Mike Meringer
David McIlwraith
Leisa Moniz*
Brian Monk
Gerard Najman
Tomas Oliva
Sig Rosenthal
Brian Stein
Richard Stern
Timothy Weisenberger *
Tena Wolf
Gary B. Yamamura
Robert Murray

Martin P. Schroeder, P.E.*,
APTA Staff Advisor for UTFS

* Non-voting member

APTA acknowledges and thanks the following organizations for contributing staff and other resources to this standards development effort:

Booz Allen Hamilton, Inc.
Cubic Transportation Systems, Inc.
ERG Transit Systems (USA)
Giesecke & Devrient Cardtech, Inc.
Los Angeles County Metropolitan Transportation Authority
Murray Associates
Port Authority of New York & New Jersey
San Francisco Bay Area Rapid Transit District
U.S. Federal Transit Administration
Scheidt & Bachmann USA, Inc.
Smart Card Marketing Solutions
Thales Transportation Systems S.A.
Three Point Consulting, Inc.
Tri-County Metropolitan Transportation District of Oregon
Washington Metropolitan Area Transit Authority

APTA acknowledges the generous support of Robert Bernard and the Port Authority of New York and New Jersey for development of the Regional Interoperable Standard for Electronic Transit Fare Payments (RIS©) on which this APTA standard has been based. APTA also acknowledges the U.S. Federal Transit Administration and the ITS Joint Program Office of the U.S. Department of Transportation for their support and guidance of this program.

Other contributors included:

Karim Aboud
David Andrews
Mauro Arteaga
Thierry d'Athis
Ed Barnette
Dianne Battilana
Al Chan
Chung Chung Tam, past chair FMC
Henk Dennenberg
Levent Eyuboglu
Christian Flurscheim
David Faust
Paula Faust
Greg Garback
Michael Guillory
Thomas Klings
Norman Kort
Michael Laezza
Etienne Lamairesse
Dominique Le Droumaquet
Gerry Lington
Rich Lobron
Darshana Patel
David Phelps
Joe Pilozzi
Ed Pollan
Henry Rosen
Traci Rozier
Rajesh Sharma
Joe Simonetti
Michael Sprague
Pete Van Antwerp
Margaret Walker
Sean Ricketson

CONTENTS

PART II - Contactless Fare Media Data Format and Interface Standard

1. Overview

1.1 Scope

This document establishes a fare media standard for transit fare collection systems and is part of a larger standard addressing other elements of fare collection standardization. Other parts of the Standard described elsewhere in Part I - Introduction and Overview, include:

Part I - Introduction and Overview (Part I)

Part III - Interface Standard between Agency Central Computer Systems and Regional Computer System (Part III)

Part IV - System Security Planning and Implementation Guidelines (Part IV)

Part V - Compliance Certification and Testing Standard (Part V)

These parts together complete the larger standard entitled, Contactless Fare Media System Standard (Standard).

1.2 Purpose

The purpose of this Part II of the Standard is to provide a consistent and uniform method for storing, retrieving and updating data from contactless fare media used in transit applications. By applying the Standard to the design of a new fare collection system or upgrade of an existing system, combined with adherence to a set of regional implementation, security and operating rules, interoperability with other compliant systems may be achieved.

1.3 System Overview

This section provides the reader with an overview of the data objects, their logical relationships, and the sequence of events in a typical fare transaction.

Figure 1 provides a high level view of the defined file structure showing the data objects within the structure. The detail behind this file structure is further described in Section 7 (File Structure).

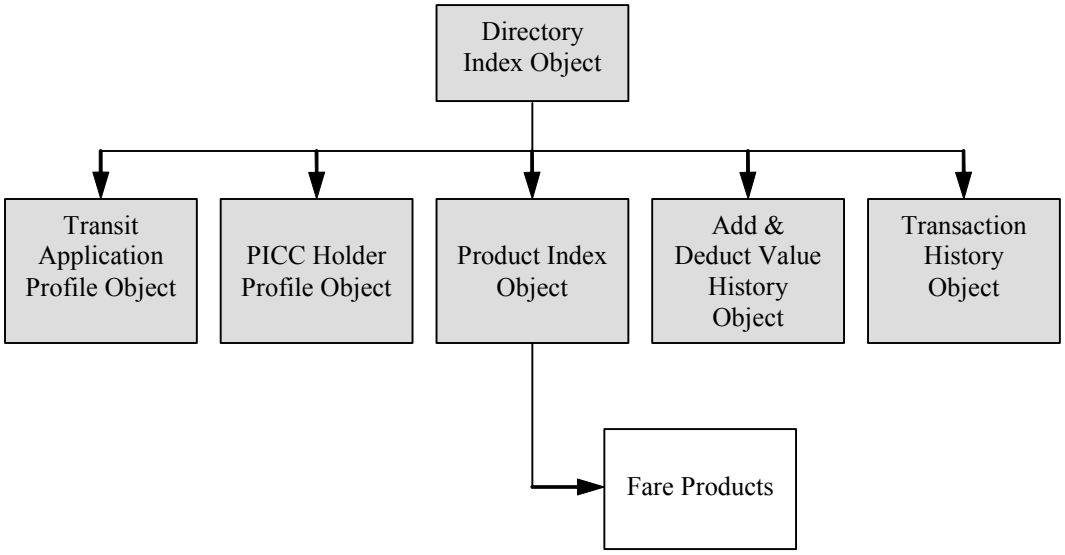


Figure 1—Generic PICC File Structure

Access to data objects should be granted through use of security keys known only between the Proximity Integrated Circuit Card (PICC) and the Card Interface Device (CID). Because of the time it takes to perform a read or write authentication or both using these keys, this document recommends the placement of many of these objects into the same file, thus allowing one read or write authentication for all grouped objects. For example, grouping the core objects as described in Section 7 (File Structure) provides for the possibility of the quickest transaction time by way of the fewest file authentication sequences.

Given these object groupings; it is possible to define a typical fare transaction that results in minimal transaction time. Actual transaction times are also subject to the specific PICC and CID hardware chosen.

1.4 Conventions

Throughout Part II, an "Rts" prefix is used before each data element name. The Rts (for Regional Transit System) is used as a general engineering practice to help identify that these data elements all belong to Part II (as opposed to Part III, which uses the same element names, but without the prefix). Using such a prefix to identify a group of data elements is a general practice among software engineers.

2. Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

— ISO 7816, Identification cards □Integrated Circuit Cards.

— ISO/IEC 14443, Identification cards □Contactless integrated circuit(s) cards □Proximity cards.

— ISO 3166, Codes for the representation of names of countries and their subdivisions.

3. Definitions

Part-I of the Standard contains definitions, abbreviations, acronyms and Standard maintenance procedures.

Each part of the Standard will be reviewed on an annual basis or as deemed necessary so as to be current with changes in technology, standard application or to address additional needs. The UTFS Task Force Financial Management Committee will coordinate and manage the updating of the Standard in accordance with procedures stipulated in Part I of the Standard.

4. PICC Objects Specifications

4.1 Transit PICC File Structure

The data architecture described in Section 7 (File Structure) is built on PICCs with a Card Operating System (COS) that supports a secure access file structure containing several core logical objects as listed in Section 4.3 (The Core Objects) (which, in turn, contain data elements). This approach facilitates International Standards Organization (ISO) 7816 Standard secure transactions while supporting full functionality of the transportation application. These objects are designed to facilitate implementation of a PICC-based flexible fare payment system which can efficiently process transactions within acceptable time limits while preventing PICC transaction tearing and resultant data corruption. The data objects are designed to support PICC platforms using Full-Featured (FF) PICCs[1].

[1]Section 4.6 (Full-Featured PICC Components) lists the PICC features that should be considered when selecting cards to support this Standard and is structured in four numbered parts. These documents are available from ISO at www.iso.org.

4.2 Full Featured PICC Requirements

Full Featured (FF) PICCs shall utilize a microprocessor-based chip with a COS, fully comply with the specifications defined in ISO/IEC 14443:[2] -4 (type A or B) and have a minimum of 2KB of useable memory for data storage. In addition, the FF PICC COS shall support, at a minimum, the following Application Protocol Data Unit (APDU) commands listed below (as defined in ISO 7816:4)[3]:

Select File
Read Binary
Update Binary
Internal Authenticate
External Authenticate
Get Challenge

NOTE—Regions implementing this Standard should understand and recognize that other industry recognized PICC products are available that meet some but not all of the requirements listed above and that may offer more attractive functionality, performance, field use, or pricing. Each Region may, at its sole option, elect to grant waivers to transit agencies within the Region to utilize one or more of such non-compliant products based on the business needs or performance requirements of the Region. In so doing, the Region must adhere to the requirements defined in Section 4.6 (FULL FEATURED (FF) PICC COMPONENTS) below (with the exception of bulleted items h and i) in order to establish a platform that supports regional interoperability.

4.3 The Core Objects

The core objects are outlined in the list below and are individually addressed in the following section of this document. Each object contains multiple data elements that define specific attributes to provide flexibility and comprehensive functionality within a PICC electronic transit application environment.

Directory Index Object (DIO)
Transit Application Profile Object (TAPO)
PICC Holder Profile Object (PHPO)
Product Index Object (PIO)
Product Object, (i.e. transit purse (T-Purse), stored value, account linked or pass)
Add & Deduct Value History Object (A&DVO)
Transaction History Object (THO)
Extension Objects

4.4 The Transit Application

The Transit Application initialized on a PICC includes the information necessary to support the transit patron profile and the patron's selected transit fare payment products. However, any application that is outside of the transportation environment does not reside within the bounds of transit application related fare payment system. For example, building security access is not part of the core functionality of the Standard. However, the Standard contains all the necessary objects and supporting data elements to be easily adapted for such applications outside of transportation fare collection. The various objects that are organized within the Standard each contain 128 bits or 16 bytes of data. The use of 16 bytes enables a wide acceptance of the Standard with relatively fast transactions over several PICC platform types.

[2] ISO/IEC 14443, Identification cards – Contactless integrated circuit(s) cards – Proximity cards is an international standard for PICCs
[3] ISO 7816, Identification cards – Integrated Circuit Cards is an international standard for smart cards and is structured in twelve numbered parts. These documents are available from ISO at www.ISO.org.

4.5 Security

Interoperability within a regional smart-card-based fare payments program cannot be achieved without establishing a regional platform for transaction, device and data security for all participating agency systems. The various methods for establishing such platforms, implementation recommendations and security terminology are described in Part IV – System Security Planning and Implementation Guidelines. Accordingly, security (while mentioned) is not extensively addressed within the text below and such references, where included herein, are provide as guidelines and are not to be considered as requirements of the Standard.

4.6 Full Featured (FF) PICC Components

The primary objectives in developing the Standard for the PICC and Proximity Coupling Device (PCD)/CID of the transit application are the following:

— Ensure the FF PICC is flexible enough to accommodate the numerous fare products of an agency or of a region or both

— Ensure the FF PICC is interoperable among multiple PCD/CIDs regardless of supplying vendors

— Ensure the FF PICC data structure and processing is based upon open standards whenever possible or applicable, enabling open sourcing of components from multiple vendors

— Ensure transit application participants can adopt new interface device technologies within the core application infrastructure

In order to accomplish interoperability the components and processing mechanisms on the PICC must be specifically defined by the agencies participating in a regional transit card program, if they are not defined in the Standard. These components and processing mechanisms include the following:

a) **PICC File Structure**

A mandatory PICC file structure is not defined in this Standard although Section 7 (File Structure) provides a suggested file structure implementation method. However, there are a number of requirements for the placement of objects (and/or their extensions) into specific files. These are described in Section 6 (File Structure Design).

b) **PICC Data Definition Mapping**

Specific PICC data objects and data elements are defined in Section 5 FF PICC Data Definition Mapping

c) **PICC Memory Requirements**

The quantity of electronic memory storage available from the PICC

Section 4.2 (Full Featured PICC Requirements) defines the minimum PICC memory requirements. Section 5 (FF PICC Data Definition Mapping) defines the size of the data elements of each object and can therefore be used to calculate the minimum FF PICC memory requirements for any given system complying with this Standard.

d) **PICC APDU Command Set**

These are the specific instructions the PICC will recognize and be able to process Section 4.2 (Full Featured PICC Requirements) defines the minimum required standard command set.

e) **PICC and PCD/CID APDU Command Sequence**

A required command sequence is not defined in this Standard.

f) **PCD/CID Order of Operations**

A required logical sequence of the PICC processing is not defined in this Standard.

NOTE—In order to achieve regional or interregional interoperability, the implementation of the Standard must also include specified components and processing mechanisms within the PICC, the CID, the broader agency system and within the regional clearing house that provide for and support the security of the regional fare payments program. These security elements are not included in the Standard.

4.7 Other PICC Types

Other types of PICCs are available to complement the implementation of Full-Featured PICCs in a fare collection program in order to provide a solution that uses only smart card based media for fare payments. These non-Full Featured (NFF) PICCs (also referenced as "Limited Use," "LU," "memory logic" and "disposable" PICCs) are generally designed to satisfy the need for a low cost fare media albeit with limited functionality and features (e.g., reduced data storage capacity, reduced durability, lower security) when compared to a typical FF PICC but must still meet the standards set forth in ISO/IEC 14443, Parts 2 and 3 (for type A or B PICCs). NFF PICCs are usually intended to replace magnetic tickets as the fare media for single rides or short duration passes and non-refillable stored value products.

5. FF PICC Data Definition Mapping

5.1 Directory Index Object

Upon presenting an enabled PICC to a PCD/CID and selecting the application, a gate or on-board PCD/CID will first access the DIO of an FF PICC, which contains a logical to physical memory mapping of the Objects stored within the PICC memory, to acquire knowledge of the PICCs contents. (Note that the maintenance of the logical to physical mapping index can be the responsibility of the respective FF PICC platform and the COS.) In the event that the selected PICC does not provide the logical to physical mapping to achieve agreed regional interoperability, a standardized DIO is provided in Table 1 Directory Index Object (DIO) as well as the DIOs position in the file and data object structure as seen in Figure 1 Generic PICC File Structure.

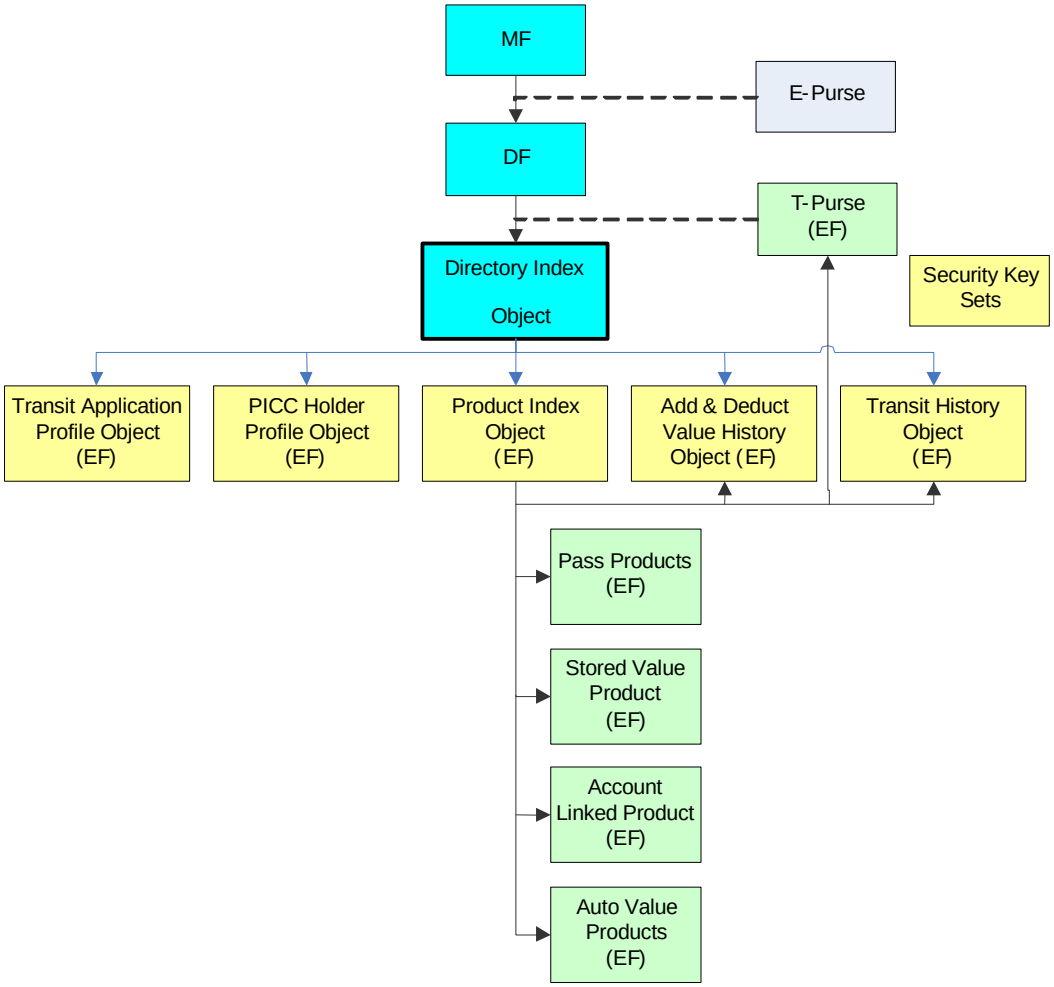


Figure 2—Transit FF PICC File Object Structure

FF Guidelines:

— The FF PICC normally contains only one DIO.

— If more than four agencies require that their files be protected by their own sets of unique keys, then a DIO Extension Object will be required.

— The DIO or an extension shall be no greater than 128 bits (16 bytes).

Table 1—Directory Index Object (DIO)

Field	Size	Values	Pos.	Description
RtsDirectoryIndexVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of four unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionStatus	1	0-1	2	0 = No DIO Extension. 1 = DIO Extension Present.
RtsTPurseExtension	1	0-1	3	Identifies if T-Purse has extension. Only valid if RtsTPurseFileID set to non-zero. 0= No T-Purse Extension (default)
RtsALROExtension	1	0-1	4	Identifier if ALRO has extension. Only valid if RtsALROFileID set to non-zero 0 = No ALRO Extension (default)
RFUDIO5-6	2	0	5-6	Reserved for future use. To be set to default value of "0."
RtsProdExtPresent	1	1	7	Indicates whether or not memory is allocated for Product Object Extensions. 0 = Product Object extensions cannot be used. 1 = All Product Objects include allocated memory for extensions.
RtsProfilePHPOExtensionID	4	0-15	8-11	Identifier for the PHPO Extension (PHPOX). 0 = no PHPOX (default) This field indicates the file number in which this extension is located.
RtsProfilePHPOExtensionQty	4	0-15	12-15	Number of PHPOX (plus one) I.e., 0000b + 1 = the number of PHPOXs. This field must be considered with RtsProfilePHPOExtensionID. This field is invalid if RtsProfilePHPOExtensionID is set to 0.

Table 1—Directory Index Object (DIO) (continued)

Field	Size	Values	Pos.	Description
RtsPIOFileID	4	0-15	16-19	Product Index Object (PIO) File Identifier The PIO is a mandatory object, and thus, this field will always be set to an appropriate value. 0 = File 0, 1 = File 1 etc.
RtsPIOObjectQty	2	0-3	20-21	Indicates the number of Product Index Object Extension (PIOX) in use. 0 = Reserved 1 = One PIO pair and one PIOX pair 2 = Reserved 3 = One PIO pair and three PIOX pairs
RtsAdd&DeductValueHistoryFileID	4	0-15	22-25	A&DVO File Identifier, e.g. 0 = File 0, 1 = File 1 etc. If 'RtsAddValueHistoryQty' = 0 then this field is ignored. All add values are recorded but the deduct function is only used in conjunction with an Action Event or Autoload such as a Directed Unload.
RtsAdd&DeductValueHistoryQty	3	0-7	26-28	Indicates the number of Add & Deduct Value History Objects: 0 = No Objects – RtsAdd&DeductValueHistoryFileID is to be ignored 1 = 2 Objects 2 = 3 Objects 3 = 4 Objects 4 = 5 Objects 5 = 6 Objects 6 = 7 Objects 7 = 8 Objects

Table 1—Directory Index Object (DIO) (continued)

Field	Size	Values	Pos.	Description
RtsTransactionHistoryFileID	4	0-15	29-32	THO File Identifier 0 = File 0, 1 = File 1 etc.
RtsTransactionHistoryQty	4	0-15	33-36	Number of THOs (plus one) I.e., 0000b + 1 = the number of THOs selected. A minimum of one THO is required.
RtsRegionalProductsFileID	4	0-15	37-40	Regional and "Other Products" selected File identification number (ID) 0 = File 0, 1 = File 1 etc.
RtsRegionalProductsQty	3	0-7	41-43	Number of Regional Product plus "Other Product" Objects, in a selected file (plus one). I.e., 0000b + 1 = the number of Regional and Other Product Objects.
RtsAgencyAProductsFileID	4	0-15	44-47	AgencyA Product Object File ID 0 = Not Used. No Agency Product Present 1 = File 1 2 = File 2, etc.
RtsAgencyAProductsQty	3	0-7	48-50	Number of Product Objects for AgencyA (plus one).). I.e., 0000b + 1 = the number of Agency1 Product Objects This field must be considered with RtsAgencyAFileID. If RtsAgencyAProductsFileID=0, then this field is invalid
RtsAgencyAID	8	1-255	51-58	Agency A's ID
RtsAgencyBProductsFileID	4	0-15	59-62	AgencyB Product Object File ID 0 = Not Used. No Agency Product Present 1 = File 1
RtsAgencyBProductQty	3	0-7	63-65	Number of Product Objects for AgencyB (plus one).), i.e., 0000b + 1 = the number of Agency2 Product Objects This field must be considered with RtsAgencyBFileID. If RtsAgencyBProductsFileID=0, then this field is invalid.

Table 1—Directory Index Object (DIO) (continued)

Field	Size	Values	Pos.	Description
RtsAgencyBID	8	1-255	66-73	Agency B's ID
RtsAgencyCProductsFileID	4	0-15	74-77	Agency C Product Object File ID 0 = Not Used. No Agency Product Present 1 = File 1 This field must be considered with RtsAgencyCFileID. If RtsAgencyCProductsFileID=0, then this field is invalid.
RtsAgencyCProductQty	3	0-7	78-80	Number of Product Objects for Agency C (plus one).)., i.e., 0000b + 1 = the number of Agency3 Product Objects
RtsAgencyCID	8	1-255	81-88	Agency C's ID
RtsAgencyDProductsFileID	4	0-15	89-92	Agency D Product Object File ID 0 = Not Used. No Agency Product Present 1 = File 1 This field must be considered with RtsAgencyDFileID. If RtsAgencyDProductsFileID=0, then this field is invalid.
RtsAgencyDProductQty	3	0-7	93-95	Number of Product Objects for Agency D (plus one).), i.e., 0000b + 1 = the number of Agency D Product Objects
RtsAgencyDID	8	1-255	96-103	Agency D's ID
RtsTPurseFileID	4	0-15	104-107	T-Purse File ID. Set to value of "0" if T-Purse not present.
RtsALROFileID	4	0-15	108-11	ALRO File ID. Set to value of "0" if ALRO not present.
CRCDIO	16	0-65535	112-127	Cyclic Redundancy Check (CRC) for error detection
Total	**128**			

NOTE 1—The format of the AgencyID fields in the DIO is the same format as those found in the PIO.

NOTE 2—The algorithm used to calculate CRC shall be CR 16 to ISO 13239 where bit 127 = Most significant and bit 112 = Least significant. The following CRC examples are provided for agencies to use in their implementation of Part II of the Standard and verification that the CRC is being calculated correctly. The 2 examples are both CRC 16 calculation results using the same sample byte stream. The polynomial used within the algorithm shall be defined by the region and used by all participating agencies within that region.

Example Data stream for the CRC is calculated (14 Byte):

Byte 0: 0×43
Byte 1: 0×52
Byte 2: 0×43
Byte 3: 0×20
Byte 4: 0×44
Byte 5: 0×45
Byte 6: 0×4D
Byte 7: 0×4F
Byte 8: 0×20
Byte 9: 0×41
Byte 10: 0×50
Byte 11: 0×54
Byte 12: 0×41
Byte 13: 0×21

1. If CRC 16 Polynomial= x16+x15+x2+1, then CRC Result: 0×6286

2. If CRC 16 Polynomial= x16+x12+x5+1, then CRC Result: 0×A16B

Start value for the CRC calculations in both cases is: 0×0000

5.2 Transit Application Profile Object

The TAPO identifies the PICCs origin, issuer, general capabilities, and limitations required by the transit framework (See Figure 3 and Table 2).

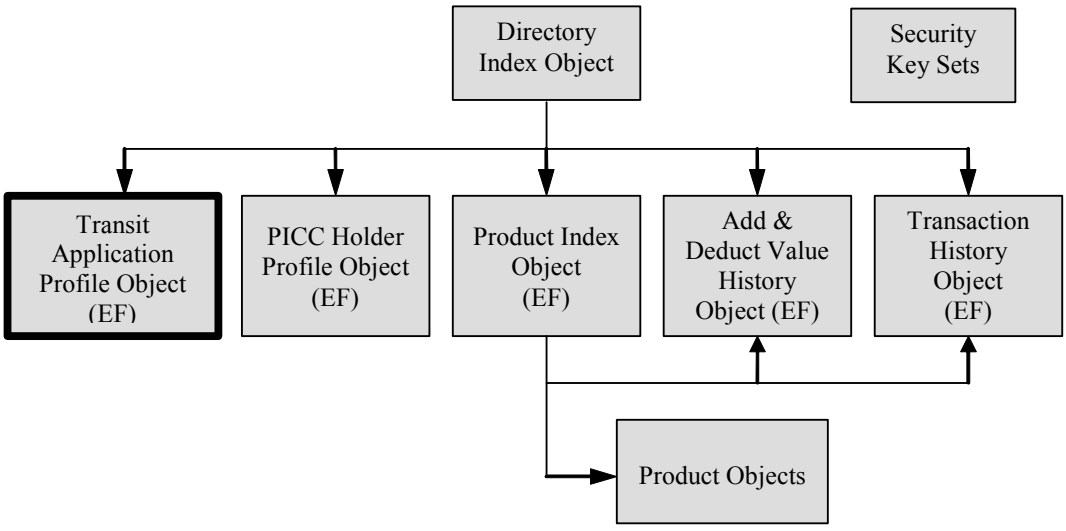


Figure 3—Transit Application Profile Object

FF Guidelines:

— The FF PICC contains only one TAPO.

— The FF PICC TAPO maintains its distinct uniqueness because it is managed by the Agency's or Region's central PICC management system.

— The TAPO is not tear proof but is normally in a read only mode.

— A TAPO shall be no greater than 128 bits (16 bytes).

The TAPO is a core object required for FF based PICC technologies.

Table 2—Transit Application Profile Object (TAPO)

Field	Size	Values	Pos.	Description
RtsTAPOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsPICCTestUse	1	0-1	2	Test or Revenue PICC indicator: 0 = Normal Operation [Revenue] 1 = Test and Maintenance
RtsOpsMaintenanceUse	2	0-3	3-4	0 = Test only mode 1 = Operations and Maintenance level-0 2 = Operations and Maintenance level-1 3 = Operations and Maintenance level-2 This field is only valid if RtsPICCTestUse is set to "1." If RtsPICCTestUse=1, and this field is not in use, then this field should be set to "0."
RFUTAPO5	1	0	5	Reserved for future use To be set to default value of "0."

Table 2—Transit Application Profile Object (TAPO) (continued)

Field	Size	Values	Pos.	Description
RtsCountryID	10	0-1023	6-15	Numeric value that identifies the country in which this PICC was issued. This element allows for values between 0 and 999. The assignment of the country code is considered fixed and permanent and consistent for all countries that recognize and adhere to the ISO 3166 standard implementing the "three Digit code" scheme, i.e.; 000 = reserved for future use 004 = Afghanistan 036 = Australia 060 = Bermuda 124 = Canada 250 = France 484 = Mexico 630 = Puerto Rico 702 = Singapore 850 = US Virgin Islands 826 = United Kingdom 840 = United States 1000-1023 Reserved By defining a country, this code is also defining the country's base currency unless there is a non-zero value populated in RtsCurrencyCode in the Stored Value and T Purse Object. The RtsCurrency Code field must always be populated.

Table 2—Transit Application Profile Object (TAPO) (continued)

Field	Size	Values	Pos.	Description
RtsRegionID	8	0-255	16-23	Numeric value that identifies the metropolitan region of a country in which this PICC was issued and intended for the majority of its use. There are 256 possible regions that can be defined on a PICC within each Country. NOTE 1 Multi-country regions may be defined with their own unique country code. NOTE 2 The assignment of the region code is considered **fixed and permanent** and consistent for all regions that recognize and adhere to the Standard. A region is defined as a grouping of transit agencies accepting the same transit PICC fare program, with the fare collection revenues being cleared and settled by an agreed upon clearinghouse system or process or both. i.e., [RtsCountryCode] = USA (840) the regional codes are as follows: 0 = reserved for future use Other values as defined by APTA or Central Entity responsible for Region ID numbering. ISO 3166 indicates region IDs and codes which may be useful in defining Region ID's. Further example; [RtsCountryCode] = United Kingdom (826): 1=London (TranSys)

Table 2—Transit Application Profile Object (TAPO) (continued)

<table>
<tr><th>Field</th><th>Size</th><th>Values</th><th>Pos.</th><th>Description</th></tr>
<tr><td>RtsIssuerID</td><td>10</td><td>0-1023</td><td>24-33</td><td>Designates the Card & Ticketing Application of the issuing Nation, State, Region or Agency, e.g.,
<table>
<tr><th>9th bit</th><th>8th bit</th><th>7-0 bits</th><th>Definition</th></tr>
<tr><td>0</td><td>0</td><td>00000000</td><td>Reserved</td></tr>
<tr><td>0</td><td>0</td><td>00000001</td><td>Agency ID</td></tr>
<tr><td>0</td><td>1</td><td>00000001</td><td>State or Regional ID</td></tr>
<tr><td>1</td><td>0</td><td>00000001</td><td>National (Gov)ID</td></tr>
<tr><td>1</td><td>1</td><td>00000001</td><td>National Institution ID</td></tr>
</table>
NOTE This field is to be considered in conjunction with the RtsCountryID and RtsRegionID fields. The codes that are specified below are applicable only for the United States (RtsCountryID=840), e.g.,

Agency use of the 7-0 bits with bits 8 and 9 set to zero.

0 = Reserved
1 = New York City Transit Authority
2 = New Jersey Transit
3 = Port Authority Trans Hudson (PATH)
4 = Port Authority AirTrain
5 = Long Island Railroad
6 = Metro-North Railroad
7 = Hudson-Bergen Light Rail
8 = New York Waterway
9 = Staten Island Ferry
10 = NFTA
11 = NYC DOT
12 = PATCO

13-63 are reserved for future transit agency IDs</td></tr>
</table>

Table 2—Transit Application Profile Object (TAPO) (continued)

Field	Size	Values	Pos.	Description
RtsIssuerID (continued)	10	0-1023	24-33	State or Regional (Non-Agency) use of the 7-0 bits with bit 8 set to a one and 9 set to zero; 0 = Reserved 1 = State of NY 2 = New York Libraries 3 = State of New Jersey 3 = State of Connecticut 4 = State of Pennsylvania 5–63 are reserved for future State or Regional IDs National Institution use of the 7-0 bits with bit 9 and 8 set to one; 0 = Reserved; 1 = Citibank; 2 = Chase Manhattan; 3 = AARP; 4 = Amtrak All other unassigned codes are reserved for future use.
RtsTransitExpirationDate	6	Date 0	34-49	The value of this field represents the Transit Application's expiration date. The expiration of the transit application is assumed to be the end of the transit day as defined by the Regional Administrator. Local Time should be defined for each device by the Regional Administrator indicated in the RtsRegionID field. Format ddmmyy 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0."
RtsTransitValidityPeriod	3	0-7	50-52	The value of this field indicates the number of years prior to the RtsTransitExpirationDate from which the Transit Application is valid. If 0, the Transit Application is valid immediately. (e.g., if this field is set to "2," and RtsTransitExpirationDate is set to "March 25, 2005," then the card would not be valid on March 25, 2003, but would become valid beginning on March 26, 2003.) 0 = Card Valid on Issue 1-7 = Card validity period in years

Table 2—Transit Application Profile Object (TAPO) (continued)

Field	Size	Values	Pos.	Description
RFUTAPO53-55	3	0-7	53-55	Reserved for future use, to be set to default value of "0."
RtsPICCManufacturerID	8	0-255	56-63	The value of this field represents the manufacturer's chip ID code. The 8 bits represent a possible 255 manufacturer chip codes. 0 is reserved or, "unknown." The codes are read by the CID in conjunction with the PICC chip serial number [UID] to ensure that vendor-to-vendor serial numbers are not repeated. The value in this field may be the only data element encoded on the PICC that guarantees the unique identity of PICCs when PICCs are procured from different manufacturers.
RtsAlternateTransitPICCID (Optional)	32	1-4,294,967,295	64-95	The value of this field may be used as the unique printed serial number assigned by the PICC manufacturer based upon instructions of the Regional Administrator. To avoid duplicate numbers, all PICC orders will be placed centrally through a single 'entity' for the requesting transit agency or retail outlet. Serial number zero is reserved. (Note that this data element has only 4 bytes of serialization where some PICCs have between 4 bytes and 7 bytes. In this case, a lookup table value would be represented by this data element. The actual serial number would be stored within the PICC silicon in a non-changeable location. This serial number may be referred to as the Alternate number. Note that a [0] in this location implies Not Present.
RtsIssuingCIDID	16	0-65,535	96-111	The PICCs issuing or encoding device's CID ID. Zero [0] is reserved. The ID must be unique for CIDs within a region.
DACTAPO or CRCTAPO Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65,535	112-127	Data authentication code (DAC) for authentication & error detection OR CRC for error detection only. Calculated on TAPO data only!
Total	**128**			

5.2.1 TAPO User's Information

5.2.1.1 RtsPICCTestUse

This data element informs the CID application that the PICC is encoded in a manner that permits its use. If a PICC is encoded to be in "Normal Operation," it is considered a patron or employee "revenue" generating PICC product. The majority of all PICCs should be in this mode once the system is "revenue ready." If a PICC is encoded to represent a "test" PICC, then it should only be implemented as a test or operations and maintenance PICC. This test mode may be associated with several usage privileges or limitations. PICCs placed into this mode should be carefully used and tracked since this mode is normally reserved for prototype system development. If a PICC is in test mode, it may be necessary for the CID application to also read optional data element [RtsOpsMaintenanceUse] to understand if the PICC is in true test mode or at what level of operation and maintenance mode.

If the RtsPICCTestUse field is set to 0 (Normal Operations), transactions that are generated shall be considered as normal Revenue Transactions. If this field is set to "1" (Test and Maintenance Mode), no transactions shall be generated by using this PICC or transactions that are generated must be distinguished from Revenue Transactions.

5.2.1.2 RtsOpsMaintenanceUse

This optional data element provides an additional level of use information. This includes normal revenue mode encoded for standard patron PICCs. The last three modes are for the purpose of operations or maintenance. A PICC issued in one of these three modes is normally issued to agency or regional employees as well as subcontractors authorized to access particular end device equipment and the internals of the devices or machines. This level of PICC privilege must be carefully administered, recorded by the device or equipment and monitored at the agency central computer since it may allow access to normally secure aspects of the PICC and CID system. The central computer and back end system should carefully monitor this type of PICC activity.

5.2.1.3 RtsCountryID

This field supports a numerical value for Country Code as defined in ISO 3166 in order to identify the PICCs country of origin. This code would normally be read in conjunction with [RtsRegionID] and [RtsIssuerID] to obtain all the information necessary to establish the origin of the PICC. Since it is possible that future regional or agency systems will accept PICCs issued from a country other than the accepting region's or agency's country of origin, this data element is necessary to direct the PICCs transaction CID and back end clearing information system to understand specific attributes of that PICC (e.g., monetary values). This data element may be used to immediately determine if the PICC should be accepted (e.g., if the accepting regional or agency system does not have the foreign PICCs country code that allows for further validation of the transaction, this would immediately terminate the transaction and limit further time being expended by the CID).

5.2.1.4 RtsRegionID

This data element provides a code that represents the particular region where the PICC was issued. This should be used in conjunction with [RtsCountryID] and [RtsIssuerID]. Further if the [RtsIssuerID] is used in conjunction with this data element, it will provide the next level of PICC issuance information, such as the actual agency or organization that issued the PICC for that region with a particular ticketing or fare application set.

5.2.1.5 RtsIssuerID

This data element informs the CID, as well as the back end processing system, of the PICCs Transit Application origin. (i.e., a PICC might be issued from a state or national organization containing its specific transit application designated by the Issuer ID code that is used by a particular region or agency for usage and acceptance).

The 8th and 9th bits of this data element are used to designate if the originating Transit Application was from a national, state, regional or agency issuer. Bits 0-7 simply provide the associated ID code of that designated issuer type.

NOTE Issuers are required to observe the conventions of the region for the transit application.

5.2.1.6 RtsTransitExpirationDate

This data element simply provides the necessary fields to set a date for the Transit Application to expire. This may also be used to signify the PICC expiration date intended for use as the Transit Application expiration. This field must be set to a valid date of expiration; however, the issuer has the option to set the expiration to a date far in the future if they do not actually desire an expiration date.

For example: This data element terminates only the Transit Application. If there were two applications on the PICC such as retail and transit, only one – say the transit application – would terminate upon reaching the expiration date encoded in this Data Element. The 16-bits used to calculate the date are structured as follows:

Bits 0-4 are 25 = 32 valid numbers (1-31)
Bits 5-8 are 24 = 12 possible numbers (1-12)
Bits 9-15 are 27 = 100 possible numbers (0-99)
Default value if field is unused is “0”
Example:
(0000101001011100) = 02/28/05

5.2.1.7 RtsPICCValidityPeriod

This data element simply provides the PICC issuer with a scheme to program the PICC with a period of validity extending from one to seven years. If a value of zero is encoded, the PICC will be valid on issue and expire on the date set by the [RtsTransitExpirationDate].

5.2.1.8 RtsPICCManufacturerID

This data element provides for an 8 bit coding of the manufacturer’s product number. There is no standardized method to access a specific manufacturer’s product within the PICCs integrated circuit. In order to minimize CID to PICC transaction time it is beneficial to determine early in the ISO/IEC 14443 acknowledgement not only the manufacturer but also the product type. Even though this data element provides the product code desired, it is past the ISO/IEC14443 negotiation and acknowledgement stage before this value can be read in the TAPO. This implies that file-0 or the core file must already be opened to read the TAPO and further this data element.

NOTE—there is a proposal at the ISO/IEC 14443 level to grant this product information early in the acknowledgement cycle. Until ISO/IEC14443 approves this new approach it is important to have an alternative method. This data element provides a reasonable solution.

5.2.1.9 RtsAlternateTransitPICCID (Optional)

This data element provides the encoding of a unique, pre-issuance serial number on the PICC, normally issued by the PICC issuer. A mirror image of the unique serial number may be provided by the PICC manufacturer and encoded into the PICC in a non-volatile memory area, before the issuance organization receives the PICC. In any case, this number must be unique and remain with the PICC for its entire life cycle. It is highly recommended that a single, responsible organization control issuing and serialization. However, the unique chip serial number [RtsPICC Manufacturer ID + UID] is the fundamental element in protecting the PICC from fraud and misuse.

5.2.1.10 RtsIssuingCIDID

This data element provides the original ID contained within the issuer's encoding equipment. This ID is assigned to the CID by the CID manufacturer and must be guaranteed to be a unique ID within the Region. The agency or regional organization must be careful that any use of encoding equipment supplied by different suppliers does not result in duplication of numbers. The CID ID must be a number from 1 to 65,535, inclusive. The organization responsible for the issuing or encoding or both of the PICC must pass this CID ID up to the back end processing center. A master list of CID IDs shall be maintained for the region and shared with each agency and the organization responsible for fraud detection or cardholder service or both. Since this ID originates in the CID, this number will be read from the CID at time of encoding and placed in this data element.

5.3 PICC Holder Profile Object

The PHPO identifies the transit patron's profile relative to their personal preferences for transit fare products and services. For the protection of the patron's privacy, personal patron data, such as name, address or other personal identification characteristics are not normally carried within the transit application. Regional, State and Federal privacy regulations and regional policy will define the level of personal information the PICC contains. Personal information requiring higher security shall be placed in the PHPOX. (See Figure 4 and Table 3).

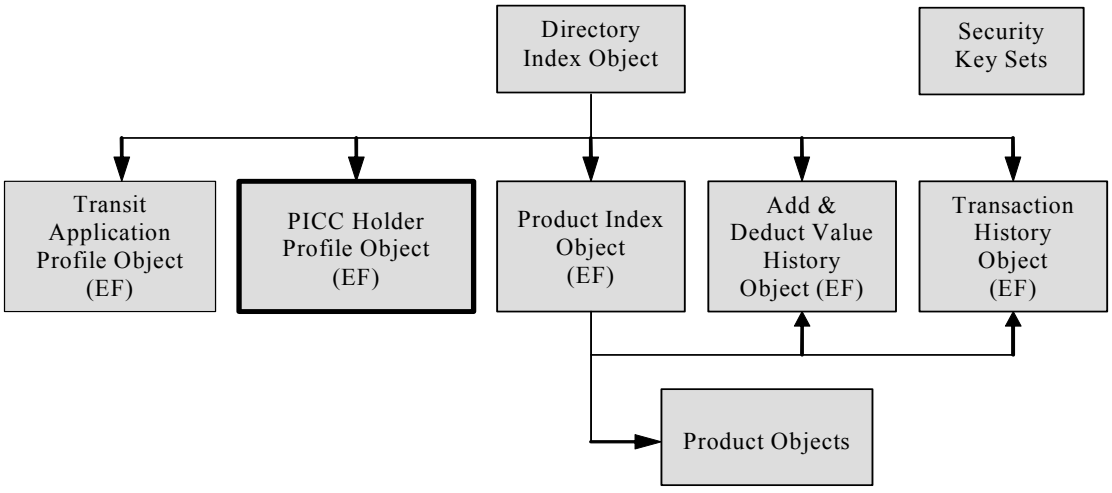


Figure 4—PICC Holder Profile Object

It is strongly recommended that this object not be altered by a CID after the PICC has been issued. If this object is required to be altered after issuance, it should only be done in a secure manner where the PICC will be guaranteed to remain in the reader field for the entire transaction to prevent tears. (e.g., manned terminals). The entire PHPO (and associated extensions) is valid or invalid based on start and expiration dates in the object. If no "valid" object is present, then the patron is assumed to have a "default" patron profile as defined by the region.

FF Guidelines:

— The FF PICC contains only one PHPO.

— An optional PHPOX must be utilized for secret or private profile protection.

— The FF PICC is managed by and maintains its distinct uniqueness via the Agency's or Region's central PICC management system.

— An FF PHPO shall be 128 bits (16 bytes).

— The PHPO is not tear protected and is normally in read only mode after initial encoding. (Modification or writing to this object is permitted after initial encoding but caution must be practiced).

Table 3—PICC Holder Profile Object (PHPO)

Field	Size	Values	Pos.	Description
RtsProfileBirthDate	16	Date ()	0-15	Birth date of the patron. To assist with identifying the patron's eligibility for discounts and recording demographics. Regional Administrator should establish implementation policy to establish oldest year that can be recorded in this field, e.g., Birthdays of > 70 years ago shall be encoded with a birth date of 70 years ago from the date of issue (if someone is 80, his birthday in year 2005 would be "encoded" to "1935"). Format ddmmyy 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0".

Table 3—PICC Holder Profile Object (PHPO) (continued)

Field	Size	Values	Pos.	Description
RtsProfileStartDate	16	Date ()	16-31	Date this profile becomes valid for use. Format ddmmyy. The start date of the Profile is assumed to be the start of the transit day as defined by the Regional Administrator. Local Time should be defined for each device by the Regional Administrator indicated in the RtsRegionID field in the TAPO. 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0."
RtsProfileExpireDate	16	Date ()	32-47	Expiration date of this patron profile and associated discount if applicable. Format ddmmyy The expiration date of the Profile is assumed to be the end of the transit day as defined by the Regional Administrator. Local Time should be defined for each device by the Regional Administrator indicated in the RtsRegionID field in the TAPO. 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0."

Table 3—PICC Holder Profile Object (PHPO) (continued)

Field	Size	Values	Pos.	Description
RtsProfileLanguage	7	0-128	48-54	To enable an automatic language of preference to display on smart PICC devices (e.g., PCD (reader), fare gate, vending machine, touch-pads at non-transit outlets, etc.). e.g., 0 = English (default, permanent standard) 1 = Spanish 2 = Chinese Traditional 3 = Chinese Simplified 4 = Korean 5 = French 6 = German 7 = Japanese 8 = Russian 9 = Hebrew 10 = Farsi 11 = Italian 12 = Polish 13 = Portuguese 14 = Vietnamese 15 = Hindi 16 = Bengali 17 = Armenian 18 = Somali 19 = Hmoob 20 – 127 = TBD
RtsUserRegistered	1	0-1	55	Indicates that the cardholder has registered prior to card issue; 0 = Non-registered or Post Issuance registered 1 = Pre-Issuance registered

Table 3—PICC Holder Profile Object (PHPO) (continued)

Field	Size	Values	Pos.	Description
RtsProfileCode	06	0-65,535	56-71	This numeric code represents a patron's specific discount or demographic profile or both if appropriate and available. These codes are extracted from the TCIP Fare Collection Document, objects section: [FC_RiderClassification_nbr] with modifications. Although standard profile codes and demographic codes should be recognized by the different participating regions for consistency, it is recognized that the criteria defining the profiles may not be consistent between regions and at times shall be region specific. As such, Profile Codes should be processed according to intra-regional and inter-regional policies or agreements, or default to adult full fare. 0-127 (Standard Patron codes) 0 = Reserved for future use 1 = Full Fare or General Fare (default, permanent full fare) 2 = Child 3 = Student 4 = Senior 5 = Youth 6 = ADA Patron 7 = Promotional 8-127 TBD 128-255 (Employee, Disabled and Special codes) 128 = Employee (full-fare) 129 = Retired Employee 130 = Public assistance patron 131 = Employee Discount A 132 = Employee Discount B 133 = Employee Discount C 134-149 = Spares 150 = Medicare 151 = Disabled with Wheelchair 152 = Disabled Blind 153 = Infant 154 = Patron with Bicycle 155-255 = TBD 256-65,535 = Reserved for future use The values that are reserved for future used could be used for "combinations" of codes (e.g., Senior, and Retired Employee), or for other codes not currently defined. These reserved codes may be determined by a central entity during the life cycle of the Standard.

Table 3—PICC Holder Profile Object (PHPO) (continued)

Field	Size	Values	Pos.	Description
RtsDepositPaid	4	0-15	72-75	Deposit Paid 0 = No deposit 1-15 = Deposit paid lookup table code The deposit value "table" is stored on the CID, or the end devices. The lookup values within the region should be maintained by the Regional Administrator.
RtsPICCHolderGender	2	0-3	76-77	PICC holder gender description 0 = Reserved 1 = Male 2 = Female 3 = Other
RFUProfile78-79	2	0-3	78-79	Reserved for Future Use. To be set to default value of "0."
RtsPICCHolderDescription	32	Not Assigned	80-111	For future assignment at the national, state or regional level.
DACPHPO Or CRCPHPO Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65,535	112-127	DAC for authentication & error detection or CRC for error detection only Calculated only for this PHPO data.
Total	**128**			

NOTE—The PICC Holder Profile Object is a core object required for FF.

5.3.1 PHPO User's Information

5.3.1.1 RtsProfileBirthDate

This data element is one entry of the PICC holder's data, which a receiving CID may use to determine ownership, special benefits and impact on fare. It is recommended that this entry exist on each PICC, but this may be made optional if the issuing operator and all system participants agree. The 16-bits are implemented as follows:

Bits 0-4 are 25 = 32 valid numbers (1-31)
Bits 5-8 are 24 = 13 valid numbers (1-12)
Bits 9-15 are 27 = 100 valid numbers (0-99)
Default value if field is unused is "0"
Example:
(0000101001011100) = 02/28/05

5.3.1.2 RtsProfileStartDate

This data element provides the starting date of a patron's specific profile. For example, a new student entering high school may be given a student profile that entitles the student to receive a discounted fare rate for travel between 7:00 A.M. and 4:00 P.M. The CID reads the start date of the profile to determine the appropriate fare-product and fare table to apply. This data element works in conjunction with the [RtsProfileExpireDate]. (See the above implementation of the 16-bits to determine a date). If the PICC is seen by a CID prior to the RtsProfileStartDate, the CID shall "assume" that the patron's profile will be the "default" profile as defined by the region. (e.g. If no valid profile is in effect, the patron is charged "Full Fare").

5.3.1.3 RtsProfileExpireDate

This data element provides for an expiration of the PICC holders profile. (I.e., if a PICC is issued to a student with benefits for a period of two (2) years from the [RtsProfileStartDate] of 28-02-04, the [RtsProfileExpireDate] is then set to 27-02-06). The CID reads the profile expiration date to determine the appropriate fare-product and fare table to apply. In the event [RtsProfileExpireDate] has been reached or exceeded, the CID would read the PICCs [RtsProfileExpireDate] and recognize no predetermined profile, so a full fare or standard fare product would then be applied. If the patron profile should expire, it will be necessary for the patron to contact customer service to reestablish the profile. If the PICC is seen by a CID after the RtsProfileExpireDate, the CID shall "assume" that the patron's profile will be the "default" profile as defined by the region. (e.g., If no valid profile is in effect, the patron is charged "Full Fare").

NOTE—If desired, the [RtsProfileExpireDate] may be set to the same date as the [RtsTransitExpirationDate].

5.3.1.4 RtsProfileLanguage

This data element enables encoding of the PICC so that it may be read by the CID application, in the very early stages of CID-to-PICC read event, to determine the language the patron prefers for transit patron interface communications (e.g., a patron presents their PICC to the CID, and the CID application by reading this data element and determining that [RtsProfileLanguage] is set to French (00000111), instructs the logic of the equipment containing the CID – such as faregate, vendor, or information terminal – to prompt any patron display or graphical user interface (GUI) to communicate with the patron in French). In the event this data element is set to a language that the end equipment does not support, the end-equipment

application defaults to English. A zero (0) or no value placed in this field is also to be understood as an English default.

5.3.1.5 RtsUserRegistered

This data element provides the necessary bit representation to the CID application to allow it to determine whether the PICC Transit Application and supporting PICC Profile were registered prior to issuance. This separate bit representing registration status facilitates rapid checking of status without the need to read any other personal information. A CID that determines the PICCs [RtsUserRegistered] field is set to a value of one (1), meaning "registered," may be used to invoke additional customer service benefits to the patron through equipment such as ticket vending equipment or other customer service terminals. A zero (0) value ("non-registered") would indicate that the RtsPICCHolderProfileDescription and PHPOX profile fields contain no usable patron information unless post-issuance changes are allowed by the Regional Administrator. This field must be encoded at the pre-issuance stage. In the event a patron decides to register with a PICC Profile, this field must be modified in advance of re-issue.

5.3.1.6 RtsProfileCode

This data element works in conjunction with the [RtsProfileStartDate] and [RtsProfileExpireDate] to establish a particular patron profile code and period of validity. In the absence of standardized industry profile codes, which do not exist as of publication of this document, non-standardized codes must be applied. In the event a code is not provided or understood by the CID application, the default must always be set to a full adult fare of (1) [0000000000000001]. Though the CID application's interpretation of the code may differ by region, the code's basic representation must not differ. If the implementation requires combining codes that are agreed upon by the region or agency, then a value between 32,768 - 65,536 is to be used to represent these combined codes (e.g., a value of 32,900 [0100000001000100] may represent the combined codes of 129 [Retired Employee] and 1 [Adult Full Fare]).

NOTE—The most significant bit signifies that the Profile Code is representing a combinational profile code. Therefore, this most significant bit functions as a point to this 32,768 - 65,536 space where these combinational profile codes are located. The lower bits represent the two combined profile codes as regionally allocated.

5.3.1.7 RtsDepositPaid

This data element enables the CID application to quickly determine whether the PICC was issued with or without a deposit paid. In the event that this field represents a value other than zero (0), this indicates that a deposit was paid. One (1) of fifteen (15) values may be encoded to provide a reference or pointer to a look-up table. This table would contain all the values used by the region or agency for deposit fees. This field must be encoded at the pre-issuance stage.

5.3.1.8 RtsPICCHolderGender

This data element contains two (2) bits that allow for three (3) gender types and a reserve bit. The value of three (3) [11] represents a status of "Other" where the patron might not want gender noted or does not agree with the title of male or female. This field must be encoded at the pre-issuance stage.

5.3.1.9 RtsPICCHolderDescription

This data element provides a field of thirty-two (32) bits that may be used to provide additional profile information as required in the future by various organizations such as the US Government. There is no

further implementation methodology of this field at this time. However, in the event more data is needed in this field, the use of a PHPOX may be implemented.

5.4 PICC Holder Profile Object Extensions

The PHPOX provides a supporting object to the PHPO that may be used store patron holder private information. (See Table 4 and Table 5)

Table 4—PICC Holder Profile Object Extension 1 (PHPOX1)

Field	Size	Values	Pos.	Description
RtsPHPOXVersionID	2		0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsCardHolderDescription	110		2-111	Used for additional PICC holder description such as full name, weight or other PICC holder required information to gain access to a site, facility, transit system or building.
CRCPHPOX	16	0-65,535	112-127	Calculated on all PHPOX data.
Total	**128**			

Table 5— PICC Holder Profile Object Extensions 2, nth (PHPOX2, nth)

Field	Size	Values	Pos.	Description
RtsCardHolderDescription	128		0-127	Used for additional PICC holder description such as full name, weight or other PICC holder required information to gain access to a site, facility, transit system or building. In addition, credit card track-two information or similar may be stored or represented in this extension object.
Total	**128**			

5.4.1 PHPOX User's Information

RtsCardHolderDescription (Optional)

This data element provides 110 bits of additional PHPO data or character based information in the first extension plus 16 bits of CRC and 128 additional bits in each of the second and third extensions that may be used to accommodate the need for detailed personal data. If this data element field is not to be used, all values must be initialized to zero (0).

5.5 Product Index Object

Located within an FF PICC, the PIO is another index type object for expediting transactions. This PIO in Figure 5 will be the index to each of the transit fare products available on a specific PICC application.

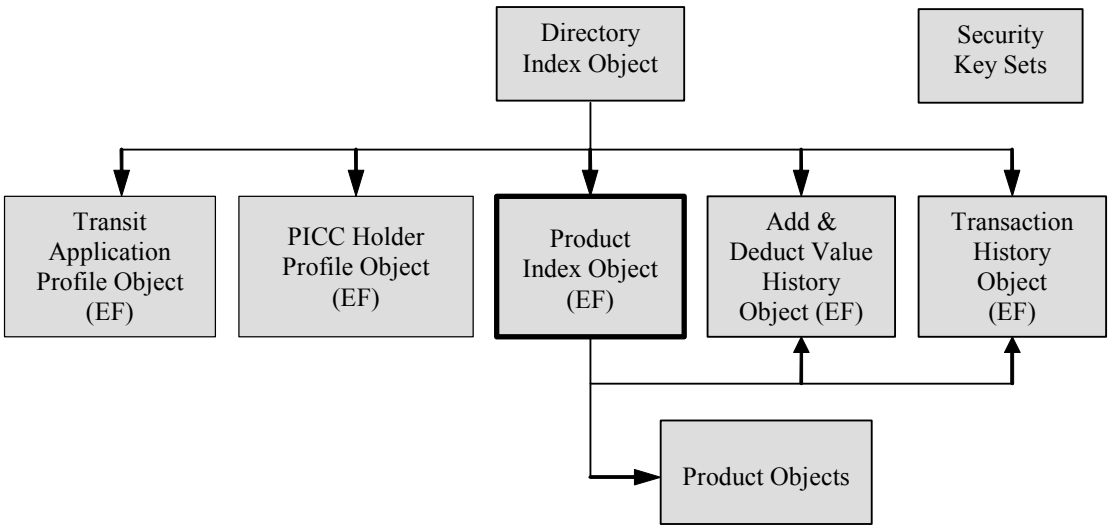


Figure 5—Product Index Object

FF Guidelines:

— The FF PICC normally contains only one PIO but can optionally have three extensions if required.

— The PIO shall be tear protected.

— The PIO is a core object required for FF PICC technologies.

The PIO will provide summarized details on the transit fare products that are currently stored on the data structure of a FF PICC, as well as the availability of these fare products, as appropriate, enabling the PCD/CID to select the most appropriate product directly, thus avoiding having to cycle through each Product Object individually. Where a product's Agency ID is valid for a transaction at a device, the PCD/CID shall assess the most appropriate product(s) based on the Product Type Code and then read the applicable product(s). Figure 6 represents the pure indexing operation of the PIO to efficiently point to the appropriate Product Object containing the actual fare product needed. (Also See Table 6)

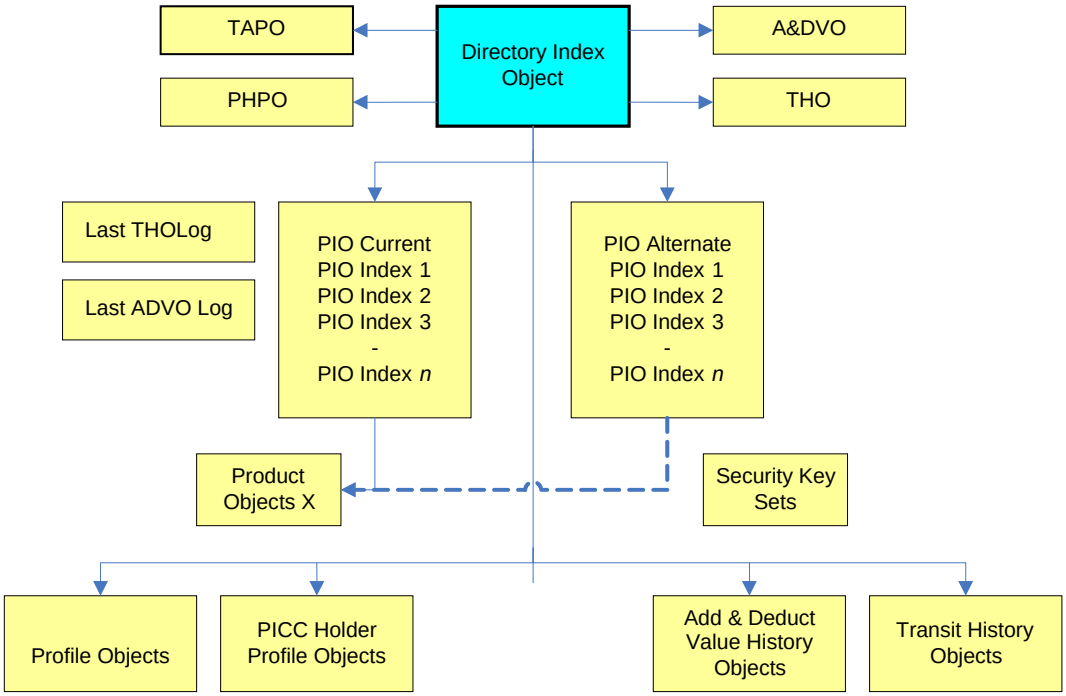


Figure 6—Product Index Object, Tear Mitigation "Mirroring"

Table 6— Product Index Object (PIO)

Field	Size	Values	Pos.	Description
RtsPIOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsHistoryObjectIndex	4	0-15	2-5	This field acts as the logical pointer to the THO for the most recent transaction in the History Log

Table 6—Product Index Object (PIO) (continued)

Field	Size	Values	Pos.	Description
RtsPICCTransSeqNumber	7	0-127	6-12	The value of this field is the PICC transaction sequence number. This PICCTransSeqNumber is updated for each completed transaction and can additionally be used for back end system transaction sorting and processing. If mirroring of the PIO is required for 'tear proofing' then the PIO with the highest Sequence Number and a correct CRC is the valid PIO. NOTE Zero becomes the highest number if the number count reaches 127 and wraps around. This field is set to "0" upon "initialization". Upon first "load," this field would then be incremented to a value of "1." The Regional Administrator should establish a policy determining how all systems must treat this field when the maximum value of 127 is exceeded. As an example, the next value for this field could always be one (1), allowing a zero (0) value to be used to identify a purse that has never been loaded. Some regions may alternatively use zero as the next value in order to simplify calculations for the total number of loads performed on a single card.
RtsAdd&DeductValueObject Index	3	0-7	13-15	This field acts as the logical pointer to the most recent A&DVO containing the most recent stored value add or the Autoload deduct (unload) in the A&DVO.
RtsLastTransactionProduct	5	1-31	16-20	This field is used to identify the primary product that is pertinent to the last THO entry. It indicates the product numbered 1-29 for the most recent transaction in the History log: 1 = Product 1 2 = Product 2 …… 29 = Product 29 30-31 = Reserved 0 = Not applicable i.e., The transaction encountered a Negative List status or a blocked or disabled PICC therefore, not processing a product.

Table 6—Product Index Object (PIO) (continued)

Field	Size	Values	Pos.	Description
RtsActionMap	8	0-255	21-28	A map of sequential bits containing 1's and 0's that may be read to achieve status of previous Autoload events. (i.e., if the posted RtsActionEventID is set at 20 and the sequential bit map contains 8 bits (bit positions 21 – 28) of which bit 27 is set to a zero, this would imply that Autoload event posted (20) minus 2 or 18 had not been retrieved). Bit 28 = Most Significant Bit, Bit 21 = Least Significant Bit
RtsActionEventID	6	0-63	29-34	This field represents the system generated Action Event Identifier for the most significant Event ID number for Autoload of this PICC. It is recorded here to prevent devices performing duplicate loads of the same product from different station or on-board devices. NOTE Pending Autoload Event Lists that are resident in the CID should be disabled if no central or back end computer activity has occurred within 72 hours.
RtsTransAppStatus	2	0-3	35-36	The value of this field represents the Negative List status of the Transit Application: 0 = PICC Transit Application is unblocked or active 1 = PICC Transit Application is negative-listed, blocked and cannot be unblocked 2 = PICC Transit Application is negative-listed, blocked but can be unblocked 3 = Reserved
RtsTPurse&SVUse	2	0-3	37-38	Indicates the use of an Agency specific Stored Value (SV) product or the Regional T-Purse: 0 = None 1 = An SV Purse was used 2 = The T Purse was used 3 = An SV Purse & the T Purse were used NOTE If an agency specific purse was used then the Agency ID is that indicated by the last Transaction History Log entry.

Table 6—Product Index Object (PIO) (continued)

Field	Size	Values	Pos.	Description
RtsPIOExtensions	1	0-1	39	Indicates the number of PIO Extensions (PIOX) in use 0 = 1 Extension 1 = 3 Extensions This field value may be changed post-issuance. However, to change from a value of "0" to a value of "1," it is necessary to "pre-allocate" space on the PICC to allow for proper extensions.
RtsALPOUse	1	0-1	40	This bit indicates if this Account Linked Product Object (ALPO) was used. 0 = Not Used 1 = Used
RFUPIO41	1	0-1	41	Reserved for future use
RtsProduct1SubIndex	3	0-7	42-44	The logical representation of this fare product's location on the PICC. This Index represents this product's position within the appropriate Product File. For agencies with their own separately keyed Product File and for the T-Purse, this will point to the position within that agency's dedicated File or the T-Purse File. For all other products, this will point to the position within the common Shared Product File. For PICCs without inherent tear proofing capability, the following provides a method for tear proofing: One position is kept vacant for a SV Product Autoload, Load or use. One position is kept vacant for a Pass Product Autoload, or an Open Dated or boardings [rides] based Pass update. When an AutoValue Product Object (AVPO) is also present, then a position will be kept vacant for its update. One position is kept vacant for an ALPO. NOTE The value in this data element is multiplied by 16 if the value of RtsProdExtPresent is 0, indicating that Product Object extensions (of 16 bytes each) are not used and 32 if the value of the RtsProdExtPresent data element is 1.

Table 6—Product Index Object (PIO) (continued)

Field	Size	Values	Pos.	Description
RtsProduct1AgencyID	8	0-255	45-52	Numeric value identifying the participating transit Agency to which this product pertains. There are 255 possible participants for each region. The assignment of the Agency IDs is unique within each region, and will be defined by the Regional Administrator. This field is always indicative of the Product Owner. 0 = A Regional Pass Product, T-Purse or Account Linked product. e.g., 1 = New York City Transit Authority 2 = New Jersey Transit 3 = Port Authority Trans Hudson (PATH)* 4 = Port Authority AirTrain 5 = Long Island Railroad 6 = Metro-North Railroad 7 = Hudson-Bergen Light Rail 8 = New York Waterway 9 = Staten Island Ferry 10 = NFTA 11 = NYC DOT 12 = PATCO 13 – 49 reserved for future use 50 = Amtrak (Regional) 51 – 255 are reserved for future use

Table 6—Product Index Object (PIO) (continued)

Field	Size	Values	Pos.	Description
RtsProduct1TypeCode	3	0-7	53-55	Product Type Code for this product. This code categorizes the transit fare products into one of eight basic types. The assignment of the product type code is considered fixed and permanent and consistent for all regions that recognize and adhere to the Standard. Valid values are: 0 = Expired 1 = Account Linked Product 2 = Time based Pass Product or Transfer 3 = Trip or Ride based Pass or Transfer Product 4 = Threshold or Recurring Autoloaded Pass Product 5 = T-Purse or SV Product 6 = Threshold or Recurring Autoloaded SV Product 7 = AutoValue Product
RtsProduct2SubIndex RtsProduct2AgencyID RtsProduct2TypeCode	14	0-7 0-255 0-7	56-69	Index, Agency ID & Type Code for Product 2. Same descriptions as for Product 1.
RtsProduct3SubIndex RtsProduct3AgencyID RtsProduct3TypeCode	14	0-7 0-255 0-7	70-83	Index, Agency ID & Type Code for Product 3. Same descriptions as for Product 1.
RtsProduct4SubIndex RtsProduct4AgencyID RtsProduct4TypeCode	14	0-7 0-255 0-7	84-97	Index, Agency ID & Type Code for Product 4. Same descriptions as for Product 1.
RtsProduct5SubIndex RtsProduct5AgencyID RtsProduct5TypeCode	14	0-7 0-255 0-7	98-111	Index, Agency ID & Type Code for Product 5. Same descriptions as for Product 1.

Table 6—Product Index Object (PIO) (continued)

Field	Size	Values	Pos.	Description
DACPIO or CRCPIO Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65,535	112-127	DAC for authentication & error detection or CRC for error detection only. Calculated on all Profile Objects.
Total	**128**			

5.5.1 PIO User's Information

5.5.1.1 RtsHistoryObjectIndex

This data element is used as a logical pointer to the THO to indicate the latest or last transaction occurrence. This pointer value must be the last data element in the transaction sequence to be modified. Since all transactions are recorded in the THO this is the master trace or history for the PICC and is therefore critical for maintaining PICC event status.

5.5.1.2 RtsPICCTransSeqNumber

This data element provides a unique number representation for each transaction. This transaction sequence number serves multiple purposes. First, the number may be used by back end systems to locate a specific transaction in a string of transactions over a given period of time. In addition, this number used in conjunction with the CRC provides a pointer to the correct PIO in the event that tear mitigation is implemented. Since the PIO may be mirrored to prevent an unrecoverable tear, it is necessary to locate the appropriate PIO, either "current" or "alternate," to achieve full recovery (see also the section on Anti-tearing in this document). A transaction sequence number may only be incremented upon the completion of a transaction. Lastly, this number may be used as a unique label for each transaction received by the back end transaction system, often referred to as a clearinghouse, to provide customer service support for use tracking as well as missing transaction recovery and fraud analysis. Since this data element (number) has a maximum of 7 bits assigned providing for only 128 unique numbers before rollover, it is important to associate this number with the transaction date and time stamp provided by the CID. The association of these two numbers ensures that each transaction has a unique identifier. Upon first issuance of the PICC, this value is set to "1."

See the following example:

PICC Seq. Transaction #	CID and time StampBack	End System Tag
10	Mar 02, 2004 13:01	010
11	Mar 02, 2004 19:53	011
027	Mar 10, 2004 08:01	027
10 (Rollover)	Mar 19, 2004 13:20	137

NOTE—In this example, since there are two transactions with the same assigned sequence number of value 10, the back end system must reference the Date and time Stamp value associated with this transaction. The back end system may elect to use the Sequence Number and Date and time Stamp to generate a unique identifier tag in the database such as 137 as seen in the table above.

5.5.1.3 RtsAdd&DeductValueObjectIndex

This data element provides a logical pointer to the most recent A&DVO. This data element may be made optional only in the event that the A&DVO is also made optional. In addition, the associated add & deduct value history object is to be used to store only "stored value" information. If this data element is implemented, the pointer logically points to the last add value transaction that occurred. E.g., if a patron loads $20.00 in stored value to a T-Purse product, the data element will point to the location where this transaction was stored. This is valuable to the CSR and the patron in the event the patron presents the CSR with an actual or perceived problem in regard to the load transaction. Since the last transaction is recorded on the PICC in this object, the CSR may gain access to this last transaction to resolve the patron's issue without the need for network-connected equipment. In the event that the patron attempted two or more value loads and reports a problem with a load previous to the last add value transaction, the CSR will need to read the A&DVO, which contains a trace of up to the last 8 load events. Any enquiry requiring recovery past the eighth entry will need to be accessed by the back end system. In any case, all A&DVO transactions shall be sent to the back end system in case they are needed at a later date. Therefore, this information may also be retrieved by the THO trace that must be sent to the back end system.

5.5.1.4 RtsLastTransactionProduct

This data element provides for five (5) bits that are used to indicate a total available product count of 32. However, the Standard is structured to support only 29 products in total, leaving value zero (0) to indicate "not applicable" (see example within the data elements description field). Values 30-31 are reserved. The 29 products are defined within the PIO and PIO Extensions 1, 2 & 3. PIO core supports 5 products while PIOX -1 supports 8 products, PIOX - 2 supports 8 products, and PIOX - 3 supports 8 more products. Note that in a typical transit environment, even in a regional implementation, the use of more than 5-6 fare products per PICC is highly unlikely. Most systems in use today support less than six fare products. In any case, this five- (5) bit data element assigns a value to identify the product that is classified as the last transaction product and is placed in the THO containing the transaction log. In the event the transit application needs to recover the last transaction, it may be necessary to know the product associated with this transaction. This data element's last value will indicate the primary product. The use of the word primary product is best explained by an example: if a stored value product is used in a manner that involves a no-fare or discounted transfer, the application will need to know the primary or core product with which the transfer was originally associated. The THO would provide this primary product information.

5.5.1.5 RtsActionMap

This data element is structured as a bit map containing eight (8) bits. It prevents the patron from missing an Action event or Autoload in the following scenario: if there are multiple Autoloads pending in the Action List with the present [RtsActionEventID] data element presently set at 20 and the Action Map bit map bit twenty-seven (27) set to zero, this would indicate that Action Event 18 was not retrieved by the patron. To clarify: Event ID 20 indicates this event being highest value loaded onto the PICC and the status of [RtsActionEventID] now reflects this retrieved event ID. However, if the patron had picked up the pending 20 event at an agency's faregate or farebox that had not been sent Action Event 18 the patron would have broken the normal sequential event list numbering (18, then 19, then 20). In this case, an Action Map is kept on the PICC so that the transit application is made aware that the normal sequence was broken and that it is likely that a skipped event is pending somewhere in the system. This may occur in a regional implementation where the central computer or back end system is fully responsible for the Action Event list generation and issuance and, therefore, is the sole entity responsible for sending these events to the participating agency end-equipment. Once the patron presents the PICC to PCD/CID that contains the Action ID 18, the PICC -- through this data element -- is aware that this Action ID event was missed and can be loaded. Once event 18 is loaded, bit map position twenty-seven (27) is set to a [1].

5.5.1.6 RtsActionEventID

This data element provides for a six (6) bit identifier for the highest value Autoload event that took place. The term Action Event is used since the term Autoload is traditionally used to refer to only the loading of value onto the PICC. Autoload (or "action event") functions may now include the unloading of a fare product or fare value, the loading of a new product or value and possibly the automatic modification of the PICCs file structure. The RtsActionEventID field is used to record these events by assigning each of them a unique identifier. Having this unique identifier, the application program can track which Autoload events were acted upon by the PICC. For example, if a patron presents the PICC to a PCD/CID and a higher value Action List event is pending that belongs to the patron, the Action Event is Autoloaded to the PICC and this data element is updated. This data element now contains the latest Action list completed so that the application is made aware that the Autoload for this Action Event was already completed preventing a duplication of the event. This data element is to be used in conjunction with [RtsActionMap]. It is recommended that any pending Action List or Autoload events be limited to 72 hours from posting to prevent excessive quantities of Action Events to be queued. This also forces the back end or clearinghouse system to manage the Action List in a timely manner. In the event that a patron has not picked-up the Autoload within the 72 hour period the system operator has the option to repost the Autoload event back onto a newly released Action List.

5.5.1.7 RtsTransAppStatus

This data element provides two (2) bits that are used by the application program to determine negative list status of the transit application, namely whether the transit application in regard to the PICC is of an unblocked or active state, negative listed or blocked permanently or negative listed and blocked but can be unblocked. This data element should be read in advance of performing any other PIO processing task to save transaction time and rapidly inform the patron by way of display that the transit application needs customer service attention before system access can be granted. The use of a [RtsTransAppStatus] set to a one (1) shall not be used unless the agency or regional operator or issuer has determined that the PICC is either stolen or corrupted or that the patron has abused the payment rules. Once set to this status level, the PICC would need to be re-initialized or destroyed and the patron re-issued a new PICC.

5.5.1.8 RtsT-Purse&SVUse

This data element provides the application with a two-bit indicator that either the SV or T-Purse is in use during this transaction. Since there is the possibility of neither being active, the data element will have a zero (0) indicating neither. In this case, an AutoValue, Pass or Account Linked product may be active.

5.5.1.9 RtsPIOExtensions

This data element provides the application with the quantity of PIO extensions that support the PIO core. Either 1 or 3 extensions are selectable. These extensions shall reside in the Core File [0] and be positioned to sequentially follow the core PIO.

5.5.1.10 RtsALPOUse

This data element provides the application with the use status of the ALPO during this current transaction. A signal bit is used to provide for the status of not used [0] or used [1].

5.5.1.11 RtsProduct1SubIndex

This data element provides the PIO with a pointer within the file where the selected product object is located and points to its actual location. The three (3) assigned bits provide for 8 possible product locations per file. However, tear proofing is required for Stored Value Autoload, Pass Product Autoload, ALPO and an AVPO product, if utilized, therefore restricting the actual number of products per file to 4. This data element is used in conjunction with [RtsProduct1TypeCode] and [RtsProduct1AgencyID] to signify what fare product type is being represented within the Product Object and to which agency the fare product belongs.

5.5.1.12 RtsProduct1AgencyID

This data element provides for an identification of the fare product's ownership. Eight (8) bits provide for 256 possible regional agency identifiers. Value (0) is reserved for a regional product or T-Purse. Values 1-255 are used to identify specific agencies or regional entities that request regional representation. Any request for agency IDs must be allocated by the Regional Administrator.

5.5.1.13 RtsProduct1TypeCode

The three (3) bits allow for eight (8) type codes. The value zero (0) shall be used to assign a given fare product an "Expired" status. When a fare product carries this status, the application shall check this value and remove this old fare product making room for a new fare product. The patron would then be able to add a new fare product since the old product was removed. Assigning a fare product with this type status code also serves as a reminder to the CID application to "clean house."

Use of a value of 4 or 6 indicates that the product was loaded as a result of a threshold or recurring Autoload. In the event the patron cancels the Autoload service, the value of this data element must be reset to 2, 3, or 5 as appropriate for the product type.

Data elements in the four rows following this element in Table 6 allow for four (4) additional fare product identifiers within this PIO. Each of these contains the same three data elements as RtsProduct1SubIndex, RtsProduct1AgencyID and RtsProduct1TypeCode. A single PIO can support up to five (5) fare product identifiers. If more fare product identifiers are needed, this may be achieved by adding PIO extensions. A total of 29 products may be supported in total.

5.6 Product Index Object Extensions

This product object is an extension of the PIO containing the same maximum of 128 bits of data. There may be up to three (3) PIOX per PIO. The implementer must exercise care in the use of PIOX since its use will increase transaction time. Table 7 below represents all of the data elements. Since this Object is an extension of the PIO, none of the supporting data elements in the PIO are needed. This extension simply provides the data space necessary to support more products beyond the five supported in the core PIO (see the PIO data elements: [RtsProduct1SubIndex], [RtsProduct1AgencyID] and [RtsProduct1TypeCode] for description).

Table 7—Product Index Object [PIO] – Extension

Field	Size	Values	Pos.	Description
RtsProduct6SubIndex RtsProduct6AgencyID RtsProduct6TypeCode	14	0-7 0-255 0-7	0-13	Index, Agency ID & Type Code for Product 6. Same descriptions as for Product 1.
RtsProduct7SubIndex RtsProduct7AgencyID RtsProduct7TypeCode	14	0-7 0-255 0-7	14-27	Index, Agency ID & Type Code for Product 7. Same descriptions as for Product 1.
RtsProduct8SubIndex RtsProduct8AgencyID RtsProduct8TypeCode	14	0-7 0-255 0-7	28-41	Index, Agency ID & Type Code for Product 8. Same descriptions as for Product 1.
RtsProduct9SubIndex RtsProduct9AgencyID RtsProduct9TypeCode	14	0-7 0-255 0-7	42-55	Index, Agency ID & Type Code for Product 9. Same descriptions as for Product 1.
RtsProduct10SubIndex RtsProduct10AgencyID RtsProduct10TypeCode	14	0-7 0-255 0-7	56-69	Index, Agency ID & Type Code for Product 10. Same descriptions as for Product A.
RtsProduct11SubIndex RtsProduct11AgencyID RtsProduct11TypeCode	14	0-7 0-255 0-7	70-83	Index, Agency ID & Type Code for Product 11. Same descriptions as for Product A.
RtsProduct12SubIndex RtsProduct12AgencyID RtsProduct12TypeCode	14	0-7 0-255 0-7	84-97	Index, Agency ID & Type Code for Product 12. Same descriptions as for Product A.
RtsProduct13SubIndex RtsProduct13AgencyID RtsProduct13TypeCode	14	0-7 0-255 0-7	98-111	Index, Agency ID & Type Code for Product 13. Same descriptions as for Product A.
RFU	16		112-127	Reserved for Future Use.

5.7 Add & Deduct Value History Object and Transaction History Object

5.7.1 General

There are two types of History Objects supported by the Standard. These include: the A&DVO and THO, both of which are stored within the PICCs data memory. The A&DVO permits up to eight Add & Deduct Value logs to be stored in the data memory before the oldest stored log is re-written by the most recent. The THO permits up to 16 logs before, the oldest stored log, is re-written by the most recent. This scheme automatically cleans or re-circulates the history log buffer so that no other housekeeping is required. Further, these logs provide tracking of the most recent transactions occurring on the PICC. The A&DVO is an optional history log. Implementation of this log capability provides for immediate patron feedback when a Customer Service Representative (CSR), is presented with the patron's PICC to identify transaction activity. The CSR can read this history log (if the PICC is functional) to inform the patron of their recent activities in the case of a transaction dispute. This history log and the THO log can be read offline since the actual transactions are stored on the PICC. The primary history that is to be stored in the A&DVO logs are value additions performed by an Autoload event or Action event, as well as a manual patron initiated vendor or point of sale (POS) value load. The pass products are typically stored in the actual associated Product Object. To prevent unnecessary value deduct transactions from using up the eight A&DVO logs, only add value transactions or reversals, or ActionEvent or Autoload "Unload" of stored value or T-Purse values are to be logged in this object. All add & deduct occurrences are stored or recorded in the THO logs.

Figure 7 represents the history objects by depicting the A&DVO and THO one past history trace or log occurrence followed by the latest transaction occurrence. The PIO provides the index pointer to the most current history position log to ensure that transaction are always made available for processing an immediate transaction or to recover past transactions. Table 8 describes the data elements in the A&DVO.

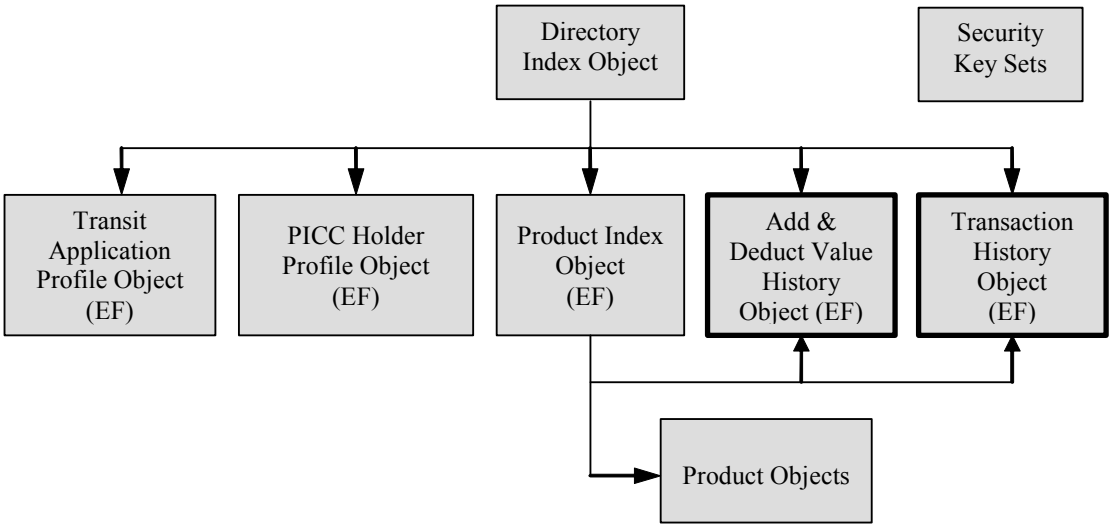


Figure 7—Add & Deduct Value & Transaction History Objects

FF Guidelines:

— The A&DVO will record up to eight of the most recent add or Autoload (Unload) value transactions.

— The A&DVO having up to eight history logs managed in a First-In-First-Out (FIFO) order.

— Each FF PICC shall have one THO with the A&DVO option implemented or not.

— The THO will contain up to sixteen transactions of ALL transaction occurrences of the most recent transactions.

— Both the A&DVO and THO can be configured to store less than the maximum number of allowed logs. (The maximum log value setting is recommended for most applications of the THO. The default of six logs is recommended for the A&DVO).

— The THO will have a minimum of six history logs.

— The A&DVO is optional.

— Each FF PICC can have one A&DVO log, this is an optional object.

Table 8—Add & Deduct Value History Object (A&DVO) (Optional)

Field	Size	Values	Pos.	Description
RtsA&DVOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsLoadType	4	0-15	2-5	Payment Type Code. Indicates the manner in which revenue was collected or returned: 0 = Reserved 1 = Cash 2 = Credit and debit 3 = Directed Autoload 4 = Threshold Load 5 = Recurring Load 6 = Set Up Threshold 7 = Set Up Recurring 8 = Set Up & Perform Threshold 9 = Set Up & Perform Recurring 10 = Cancel Threshold 11 = Cancel Recurring 12 = Value Based Coupon 13 = Trade-in 14 = Check 15 = Reserved

Table 8—Add & Deduct Value History Object (A&DVO) (Optional) (continued)

Field	Size	Values	Pos.	Description
RtsValueExpires	1	0-1	6	Expiry indicator. Indicates that this is the load of a single load Stored Value product. A value of "0" indicates that the applicable purse product does not expire. A value of "1" indicates that the purse product does expire. 0 = Normal 1 = Expires
RFUA&DVO7	1	0	7	Reserved for future use.
RtsAgencyID	8	0-255	8-15	For Agency specific SV purse this is set to the relevant Agency ID. The Agency ID for the Regional T-Purse is set to 0.
RtsDate	16	Date () or zero	16-31	Transaction Date. Format ddmmyy. 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0".
RtsTime	11	0-1439	32-42	Transaction Time in minutes past midnight (24:00 hour's clock).
RFUA&DVO43	1	0	43	Reserved for future use.
RtsRecurringAutoloadType	4	0-15	44-47	Used to differentiate T-Purse or SV Recurring Autoload types: 0 = Reserved 1 = Weekly 2 = Monthly 3 = Bi-Annual [6 monthly]4 = Annual 5-7 Reserved Recurring autoloads are performed as defined by Regional Policy. (e.g., A region's policy could be to perform weekly autoloads on Tuesdays).
RtsSVTransactionValue	15	0-32,767	48-62	Value Added or Deducted inclusive of any bonus for Cash, Bank Card, Directed Autoload or Threshold Autoload Transactions. This data element includes any bonus which was added or included in the transaction. The amount paid can be calculated with this data element in conjunction with external data.

Table 8—Add & Deduct Value History Object (A&DVO) (Optional) (continued)

Field	Size	Values	Pos.	Description
RtsSVTransactionNegative	1	0-1	63	0 = Positive 1 = Negative
RtsSVTransSeqNumber	7	0-127	64-70	The value of this field is the transaction sequence number of the stored value object or T-purse object that was used. This TransSeqNumber is updated for each completed transaction involving the product in question, and can additionally be used for back end system transaction sorting & processing. This field is set to "0" upon "initialization." Upon first "load," this field would then be incremented to a value of "1." The Regional Administrator should establish a policy determining how all systems must treat this field when the maximum value of 127 is exceeded. As an example, the next value for this field could always be one (1), allowing a zero (0) value to be used to identify a purse that has never been loaded. Some regions may alternatively use zero as the next value in order to simplify calculations for the total number of loads performed on a single card.
RtsLocationID	16	0-65,535	71-86	The value of this field represents the unique location of the device within the regional system. This field should be maintained by the Regional Administrator to avoid duplication of values within a region. Applicable field value ranges from 1 to 65,535. Zero (0) is reserved.
RFUA&DVO87-88	2		87-88	Reserved for future use.
RtsCIDTransactionNumber	7	0-127	89-95	This data element provides a unique number for each transaction that occurs with the CID. This data element will be the 7 least significant bits of the CID Transaction sequence number.
RtsCIDID	16	0-65,535	96-111	Issuing Machine CID ID. Used to identify the transaction encoding equipment. CID IDs are administered by Regional Administrator and should be unique within the region.

Table 8—Add & Deduct Value History Object (A&DVO) (Optional) (continued)

Field	Size	Values	Pos.	Description
DACA&DVO or CRCA&DVO	16	0-65,535	112-127	DAC for authentication & error detection Or CRC for error detection only
Total	**128**			

5.7.2 A&DVO User's Information

5.7.2.1 RtsLoadType

This data element indicates the method in which the revenue was collected or returned. The purpose of this data element is to provide a trace or history of the transaction event that occurred for the add value load type. The add value transaction could be an Autoload, patron directed load or customer service unload transaction. In each of these cases the appropriate load type representation in this data element should be used to help identify the transaction in case a patron dispute or inquiry occurs. Four bits are allocated, permitting fifteen (15) different payment type codes with [0, 15] held in reserve.

5.7.2.2 RtsValueExpires

This data element allows for an expiration date to be set for the stored value load implying that the actual load event is for a single or one-time load. The actual Stored Value & T-Purse Product Object (SV&TPPO) makes use of the data element [RtsExpRecurDate] to determine when the single load event expires. A recurring Autoload event for an expiring purse, would also be signified by [1] being set. If a [0] bit is set, the stored value load event is not for an expiring purse.

5.7.2.3 RtsAgencyID

This data element provides the assigned ID of the agency that is permanent and consistent within the region for the stored value product. In the case of a T-Purse that is only assigned by the regional operator, the value in this data element is set to [0]. This data element corresponds with the SV&TPPOs data element [RtsProduct1AgencyID] where [0] is regional and [X] represents agency IDs.

5.7.2.4 RtsDate

This data element indicates the date the add value or deduct transaction occurred. This is part of the time and date stamp issued with every transaction to ensure that each transaction can be uniquely identified. This data element must be used in conjunction with the data element [RtsTime]. Their combined information provides all the necessary usage time and date stamp information. Sixteen bits are allocated where 2 digits per field are used. Since the year field uses only 2 digits where four may be applied, care needs be taken in representing only dates that are of the years 2000 to 2099 where "00" is the year 2000 and it should be noted that for dates indicating the past, "00" would indicate the year 2000. In the event all zeroes are placed in the digits, this will indicate that there is no date set. In this case, the CID application should immediately flag this absence of date as potential fraudulent activity. The 16-bits are implemented as follows:

Bits 0-4 are 25 = 32 possible numbers (i.e., day 28 is {11100})
Bits 5-8 are 24 = 16 possible numbers (i.e., month 02 is {0010})
Bits 9-15 are 27 = 128 possible numbers (i.e., year 05 is {0000101})

Or

(0000101001011100) = 02/28/05

5.7.2.5 RtsTime

This data element provides the time stamp of the "add" or "deduct" value transaction with a granularity of one minute. In the event two add value or deduct transactions were to occur within the same minute of time, the transaction time and date stamp would not be unique. The eleven bits assigned to this data element allow for 2048 minutes of which only 1439 are needed. The minutes are managed from the start of 24:00 hours so for example, minute 648 would represent 10:48 hours or 10:48 am.

5.7.2.6 RtsRecurringAutoloadType

This data element is used as a high-level indicator to differentiate the stored value or T-Purse recurring load classifications. Four bits are used to indicate the type of recurring Autoload event, i.e., weekly or monthly. The reserved values may be used to indicate other Autoload event types such as bi-annual or annual.

5.7.2.7 RtsSVTransaction

This data element provides for the actual value of the Stored Value or T-Purse load or deduction transaction. Fifteen bits are allocated for this data element that may be represented as a unit value such as: one-cent, one nickel, one-dime, one-quarter or one-dollar. For example, if one-nickel unit granularity were to be selected, the 15 bits would permit an actual value representation of $0.00 to $1,638.20. If a value greater than this were to be required, then the application should use a granularity of greater than a nickel. Any value added to or deducted from the SV or T-Purse Product Object must be recorded in this data element.

5.7.2.8 RtsSVTransactionNegative

This data element indicates whether the value associated with the add value product transaction is a negative or positive value. This data element derives its status from the sign of the value being added to the SV&TPPO by the load vendor or via the Autoload method. The application must acknowledge this negative or positive sign bit indication to correctly track patron activity. Since all value added or subtracted from the SV&TPPO may be recorded in the A&DVO log this data element provides an immediate indication of the transaction arithmetic sign.

5.7.2.9 RtsSVTransSeqNumber

This data element provides a unique sequence number representation for each TP or SV Product's transaction. This transaction sequence number may be used to locate a specific purse transaction in a string of transactions for that purse over a given period of time. The transaction sequence number may only be incremented upon the completion of a transaction. This number may be used as a unique label for each Purse product's transaction received by the back end transaction system, often referred to as a clearinghouse, to provide customer service support for use tracking as well as missing transaction discovery and fraud analysis.

5.7.2.10 RtsLocationID

This data element provides for 16-bit location identifier of the CID. Each agency or regional organization shall assign a unique identifier for each of its locations. This data element could also serve as a pointer into a lookup table giving the physical description of where the CID is located. For example, [RtsLocationID] has a (0000000001101000) loaded in its field representing location ID 104. This ID of I04 could point via a look-up table in a back end file that defines location ID 104 as Journal Square located in New Jersey. The transaction event stored in the THO would not only indicate that a PICC with a given serial number belonging to agency "X" with product "Y" was loaded, but where and by which CID.

Location IDs should be maintained by the Regional Administrator to avoid duplication of values within a region.

5.7.2.11 RtsCIDTransactionNumber

This data element provides a unique number for each transaction that occurs with the CID. The [RtsCIDTransactionNumber] is used by the transaction acquirer or agency of use to check for missing transactions at their devices in a similar way to that which the RtsSVTransSeqNumber is used by a product issuer or owner to track missing transactions. This data element is used in conjunction with [RtsCIDID] to track the unique transaction number to the actual CID performing the transaction. Each transaction data packet that is completed and sent up to the back end system for accounting and settlement shall contain both [RtsCIDID] and this data element for tracking purposes. It is imperative that the CID Transaction Number is not corrupted due to power loss.

If a Product Sale or Autoload Transaction from a Product Retailer is lost or delayed due to off line operation of a device or an attempted fraud, the [RtsCIDID] & [RtsCIDTransactionNumber] are both stored on the PICC such that they can be transmitted to the Product Owner on the next use of the PICC or first use of the Product (or both) to which they pertain. This enables the product owner to reconcile missing transactions with the Product Retailer using the actual 'Receipt Number' for the sales transaction. It also potentially provides Product Retailer Sales Acquirers (that may acquire 3rd Party Sales transactions) with vital tracking or fraud analysis feedback regarding any lost or missing sales transactions that they themselves do not receive.

5.7.2.12 RtsCIDID

This data element provides for unique serialization of each CID within the agency or region. Sixteen bits are provided to allow for 65,535 unique ID's. Since the highest known CID count in any one region is estimated to be less than 30,000 this data element's ID count should be sufficient to cover all regional possibilities. Anytime a transaction occurs, this data element must be read and sent up to the back end system as part of the packaged data. This is especially important each and every time a CID is used to load value onto a PICC. The CID ID in conjunction with the actual transaction data provides the necessary information to help identify the transaction and to prevent fraud. Each CID must have a unique number assigned prior to or at the time of field deployment. A responsible implementation method would involve assigning a correlation between the CID ID and the actual physical location of the CIDs deployment through a lookup table or other method. It is important to know not only which CID performed the transaction but where the CID is physically located.

5.8 Transaction History Object

Table 9 described the data elements in the THO. (See Section 5.7 for overview)

Table 9—Transaction History Object (THO)

Field	Size	Values	Pos.	Description
RtsTHOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionStatus	1	0-1	2	0 = No THO Extension (THOX) 1 = THOX required NOTE□ The THOX position must precede the core THO.
RtsTransactionType	4	0-15	3-6	Denotes the type of Transaction: 0 = Reserved 1 = Load 2 = Product Blocked 3 = Product Un-Blocked 4 = Validation or Deduction 5 = Validation or Deduction Date and time Override 6 = Reserved 7 = Configuration Change 8 = Previous transaction Undone 9 = Reserved 10 = Reserved 11 = [RtsTransAppStatus] Change 12 = Unload 13 = Reserved 14 = Reserved 15 = Out of Region T-Purse Use

Table 9—Transaction History Object (THO) (continued)

Field	Size	Values	Pos.	Description
RtsIn/Out	1	0 or 1	7	This field could indicate either WHERE the transaction was located (paid, unpaid area), or an ACTION (entry or exit). The manner in which this flag is used must be determined by a region. Must be used in conjunction with transaction type. 0 = Out 1 = In
RtsProductType Or RtsRegionID	8	0-255	8-15	Product Type for in Region Product Use. Or Region ID for out of Region T-Purse Use.
RtsAgencyID	8	0-255	16-23	This field represents the entity performing the transaction. (e.g., Service Provider or Product Retailer, etc.) 0-254 As determined by the Regional Administrator 255=Reserved for "See lookup table"
RtsLocationID	16	1-65,535	24-39	This data element provides for 16-bit location identifier of the CID. Location IDs shall be assigned by each Agency to each CID within its system and a master list shall be maintained by the Regional Administrator to avoid duplication of values within a region.
RtsDateStamp	16	Date ()	40-55	Date of transaction. Format: ddmmyy 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0" Local Time should be defined for each device by the Regional Administrator.
RtsTimeStamp	11	Time 0	56-66	Time of transaction. Time in minutes past midnight (24:00 hours) when transaction occurred. Format: mmm [0 – 1339] Local Time should be defined for each device by the Regional Administrator.

Table 9—Transaction History Object (THO) (continued)

Field	Size	Values	Pos.	Description
RtsTransactionLinked	2	0-3	67-68	The next fare calculation is: 0 = Not Linked to any previous THO record 1 = Linked to a previous validation/deduct THO record other than the most recent 2 = Linked to the most recent validation or deduction THO record 0 = Reserved
RtsTransferStartTime	11	0-1439	69-79	Transfer Start Time for the journey. Used to determine transfer validity. Time in minutes past midnight (24:00 hours). Set to Time Of Use if not applicable or for start of new journey. The value of this field should be set to the same value as RtsTimeStamp if this field is not applicable.
RtsTransValue	15	0-32,767	80-94	Value of the SV or T Purse transaction, where applicable. 0 = not used.
RtsTransValueSign	1	0-1	95	Value designates a positive or negative Transaction Value: 0 = Positive (Add) 1 = Negative (Deduct)
RtsTransferCode	8	0-255	96-103	Transfer Service Code
RtsSpecial (Optional)	8	0-2585	104-111	Bits reserved for Agency specific usage. I.e., a single bit or a set of bits may be used for one or more of the following: - Bus Number Hash - Transfer Count 0 = Not Used and reserved. If this field is not used, it should be set to 0.
DACTHO or CRCTHO DAC or Cyclic Redundancy Check (CRC)	16		112-127	DAC for authentication & error detection Or CRC to verify integrity of this object.
Total	**128**			

NOTE—The THO is a core object required for FF PICCs.

5.8.1 Transaction History Object User's Information

5.8.1.1 RtsExtensionStatus

This data element provides a single bit status to indicate whether a THOX exists. In order to correctly interpret the transaction log, the PCD/CID must be made aware that the log contains more data than the core 16 bytes.

5.8.1.2 RtsTransactionType

This data element provides for a 4-bit field giving 16 possible transaction type codes to help identify in the History Log the type of transaction product that was stored and its association with a load or use transaction type. For example, a THO slot that contains identifier [1] (Pass Product Load), informs the application attempting to recover or operate from this transaction (the last transaction) that the transaction was not only a Pass Product but also that it was a Pass Product that was loaded.

5.8.1.3 RtsIn/Out

This data element provides the application with a single bit status that may be used to verify whether the transaction occurred within or outside of the paid or closed area or upon entering or exiting the system. .In the case of a Pass product being loaded at a convenience store that exists outside of the paid area or closed system, the bit would be [0]. This may be used by the back end system to determine quickly that the transaction is not to be found in one of the controlled regional agency areas.

5.8.1.4 RtsProductType or RtsRegionID

This data element normally provides a record of the product type used (i.e., the ProductType of the appropriate Product Object). However, in the case of an interregional product (out of region) such as a T-Purse that was issued by another region but accepted in this region, the RtsRegionID is recorded instead of RtsProductType. This occurs when the RtsTransactionType field has a value of "15."

5.8.1.5 RtsAgencyID

This data element provides a record of the agency ID taken from the PIO. Since all transactions are stored in the THO, it is necessary to have an agency ID so that the product may be associated with the agency of use and or issue. The 8-bit field provides for 256 possible agency IDs. This is most important when attempting to recover or determine where a PICC was corrupted or where a load event occurred and what agency system completed the transaction.

5.8.1.6 RtsLocationID

This data element provides for 16-bit location identifier of the CID.

Location IDs shall be assigned by each agency to each CID within its system and a master list shall be maintained by the Regional Administrator to avoid duplication of values within a region.

5.8.1.7 RtsDateStamp

This data element provides a record of when the actual transaction event took place by date. It is extremely important that the transaction event date be correctly stamped and recorded. This data element uniquely identifies the transaction occurrence. The data element must be used in conjunction with [RtsTimeStamp] to fully define the transaction event uniquely. The combined use of these two data elements must occur at the completion of the transaction performed at the CID. The two data elements are sent up to the back end computer along with other data recorded in the THO to create the transaction record. The 16-bits are implemented as follows to form a date:

Bits 0-4 are 25 = 32 possible numbers (i.e., day 28 is {11100})
Bits 5-8 are 24 = 16 possible numbers (i.e., month 02 is {0010})
Bits 9-15 are 27 = 128 possible numbers (i.e., year 05 is {0000101})
Or
(0000101001011100) = 02/28/05

5.8.1.8 RtsTimeStamp

This data element provides a record of when the actual transaction event took place by time. It is extremely important that the transaction event time be correctly stamped and recorded. This data element uniquely identifies the transaction occurrence. The data element must be used in conjunction with [RtsDateStamp] to fully define the transaction event uniquely. The combined use of these two data elements must occur at the completion of the transaction performed at the CID. The two data elements are sent up to the back end along with other data in this Product Object to create the transaction record. The structure of this element is used in the following method:

Bits 56-66, = 0 to 211 –1 = 0 - 2047 but represents 0 – 1439 possible minutes past midnight
Or
Example: (01011010000) = 720 minutes past 24:00 hours

5.8.1.9 RtsTransactionLinked

This data element uses 2-bits giving four possible status identifiers as to the last transaction having links to the previous transaction. This would be commonly used when a SV&TPPO is used and transfers are granted. The use of the transfer would be linked to the previous transaction by this data element. If the previous transaction were associated with a transfer that had been interrupted by another transaction a [2] or 10 binary would be written in this field to signify that more information is necessary to recover the previous transaction in association with previously granted transfers (see Section 5.7 on Transfers for more details).

5.8.1.10 RtsTransferStartTime

This data element provides the recorded start of a transfer or start of journey. It makes use of 11-bits to count forward from 24:00 hours to establish the transfer start time. This data element must be used in conjunction with [RtsDateStamp] to determine when the transfer started and on what day or 24 hour clock (see [RtsTimeStamp] to determine how the 11-bits are applied for a minute counter).

5.8.1.11 RtsTransValue

This data element provides the value in the last SV&TPPO transaction. The 15-bits are used to establish 32,768 units of value. The application program determines the actual worth of each unit value, e.g., if a unit value is worth $0.10 then the Stored Value transaction could contain a maximum of $3,276.70. If transaction were a SV&TPPO load, this unit value would also be recorded in the A&DVO.

5.8.1.12 RtsTransValueSign

This data element uses a single bit to provide status as to the value of the Stored Value or T-Purse transaction in terms of negative or positive value.

5.8.1.13 RtsTransferCode

This data element provides an 8-bit transfer code that is used in conjunction with a Product Object associated with issue of a transfer or multiple transfers. In order to track the transfer when it is interrupted from the previously associated primary fare product or Product Object, this transfer code assigns a unique ID that allows the transfer to be associated with its original Product Object (see Section 5.7 for more details on transfers and Transfer codes).

5.8.1.14 RtsSpecial (Optional)

This data element provides 8-bits to be used by the agency or regional organization to define and track special occurrences. This data element is considered optional.

5.9 Transaction History Object Extension

Table 10 describes the elements in the transaction history object extension, (THOX). (See Section 5.7 for overview)

Table 10—Transaction History Object Extension

Field	Size	Values	Pos.	Description
RtsTHOXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).

Table 10—Transaction History Object Extension (continued)

Field	Size	Values	Pos.	Description
RtsExtensionType	6	0-63	2-7	Identifies the type of THOX 0 = Group SV Travel [This example] 1-63 = Reserved for Future Use This field would be defined when an applicable field is determined by the Standard body. Currently, the Standard only defines RtsExtensiontype=0. Additional values and extension types will be defined as the Standard evolves. The definition of the extension type will define the data elements within this extension.
RFUTHOX8-15	8	0-255	8-15	Reserved for future use.
RtsProfileCodeRider1			16-23	These values are to be the same values and representations as indicated in the RtsProfileCode field in the PHPO.
RtsTransCodeRider1	8	0-255	24-31	Transfer Service Code associated with this use. Used to determine next use transfer charge.
RtsProfileCodeRider2	8	0-255	32-39	See above
RtsTransCodeRider2	8	0-255	40-47	See above
RtsProfileCodeRider3	8	0-255	48-55	See above
RtsTransCodeRider3	8	0-255	56-63	See above
RtsProfileCodeRider4	8	0-255	64-71	See above
RtsTransCodeRider4	8	0-255	72-79	See above
RtsProfileCodeRider5	8	0-255	80-87	See above
RtsTransCodeRider5	8	0-255	88-95	See above
RtsProfileCodeRider	8	0-255	96-103	See above
RtsTransCodeRider6	8	0-255	104-111	See above
RtsProfileCodeRider7	8	0-255	112-119	See above
RtsTransCodeRider7	8	0-255	120-127	See above
Total	**128**			

5.9.1 Transaction History Object Extension User's Information

(See THO User's Information)

5.10 Product Objects

5.10.1 General

Product Objects define the transit pass, value, Account Linked and/or AutoValue based products of the respective participating agencies. This object model, as shown in Figure 8 accommodate any fare products that are pass, value, account linked, or patron reward based. Each FF Product Object may be stored in a separate, Agency-specific file protected by a unique key set or the Object may be stored in a common file and protected with a shared key set. Two examples of special Product Object functionality being supported are given below. There are other functional considerations provided for in the Product Objects, therefore this is not a complete representation. The implementer must extract from this document any additional special functionality that may address a particular agency or regional implementation.

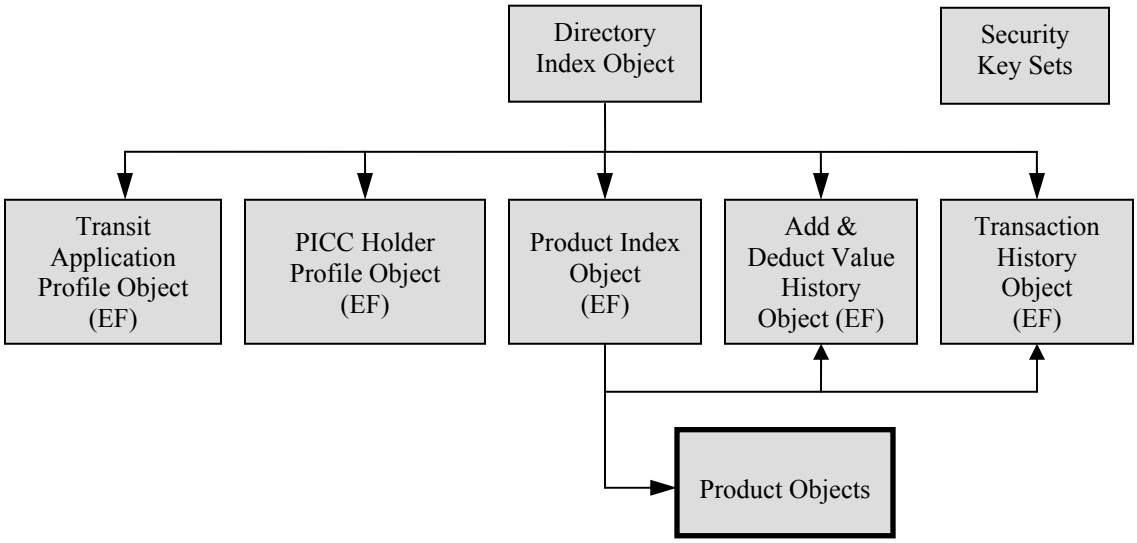


Figure 8—Product Objects

FF Guidelines:

— A given transit fare product is represented by a single Product Object.

— There can be one or more fare products per FF PICC. This is constrained by the memory limits of the FF PICC and maximum addressing of the PIO Extensions.

— Each transit fare product can have a minimum of one Key Set (Read and Write or a combined read and write key), if required.

— All products can be automatically reloaded by a PCD/CID if ActionEvent/Autoload subscription services are enabled by the PICC holder.

— There are 256 distinct Product Types available to each Agency (E.g., 253 pass based products, one T-Purse, one account linked and one AutoValue based product. However, the PIO supports only 29 products per transit application.

5.10.2 Autoload/ActionEvent (Special functionality)

A transit patron who subscribes to the ActionEvent/Autoload service via a back office administrative process will automatically have their designated fare transit products or purse products re-loaded on a predetermined schedule or at a predetermined threshold. The patron may define the frequency of these value loads, by either date or request threshold Autoloads as part of the ActionEvent. It is left to the responsible agency to set the threshold value amount limits. For example, the patron can establish a time frame in order to re-load a fare transit product (e.g., monthly pass on the first day of the month), or a value threshold (e.g., a value that becomes less than $5.00). The PICC will contain the ActionEvent and Autoload indicators. In addition, an ActionEvent may be used to unload a previously loaded value or product from the PICC transit application.

5.10.3 Negative Balance (Special Functionality)

Value products are stored value or highly secured T-Purse based fare products for transit agencies or regional operators. One aspect of these value products is the ability to allow for negative balances. As an example, if the patron has only a stored value or T-Purse product that contains a balance of $1.00 and he or she does not have the Autoload subscription service enabled, a fare of $1.50 or more would normally stop that patron from passing through the gate. This would be an inconvenience to the patron and a transit passenger throughput issue, so the system feature of Negative Balance will allow temporary negative balances. [Depending on regional fare policies and operating rules, a PICC deposit may offset some limited financial exposure to the transit agencies when incurring a negative balance.] If the balance remains negative, the next time the patron tries to use the stored value on the PICC: the system will decline the transaction and may possibly temporarily disable the PICC transit application. The customer must add value to the negative balance and provide sufficient positive value for future use. Implementation of this feature depends on specific agency fare policies.

5.11 Pass and Transfer Product Objects

The use of a single Product Object description that contains the required information or functionality and data representing either a pass or transfer product is defined by Table 11. A pass product can be used as either an agency pass fare product or a regional pass product.

Table 11 —Pass and Transfer Product Objects

Field	Size	Values	Pos.	Description
RtsP&TProductVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionStatus	1	0-1	2	0 = No P&TPO Extension (P&TPOX) 1 = P&TPOX required NOTE The extension(s) position immediately follows the core P&TPO.
RtsAutoloadSubscribed	2	0-3	3-4	Autoload Selection where 0 = Not Subscribed 1 = Subscribed 2 = User Defined (Optional) 3 = User Defined (Optional)
RtsPaymentType	2	0-3	5-6	Payment Type Code. Indicates the manner in which revenue was originally collected for the purchase of the Product: 0 = Cash or T-Purse [Refundable] 1 = Credit and debit or Stored Value [*non-refundable or refundable] 2 = Directed Autoload 3 = Subscribed Autoload * Policy would determine if this Payment type should be refundable or non-refundable.

Table 11—Pass and Transfer Product Objects (continued)

Field	Size	Values	Pos.	Description
RtsRenewedInAdvance	1	0-1	7	Indicates that this Product has been renewed in advance of its existing expiry Date 0=Not renewed in advance 1=Renewed in advance NOTE The actual renewal invocation will only take place after the existing product has expired. The renewal will invoke a new product written with a new expiry date and time.
RtsLocationEncodingType	1	0-1	8	Describes the type of location validity encoding depicted by [RtsLocationEncoding] data element. 0 = Agency's encoding Format Type 0. i.e., Sector or Route Coding 1 = Agency's encoding Format Type 1. I.e., Point to Point Coding
RtsExpDate	16	Date () or zero	9-24	Expiry date for the Product where: Bits – Denotation 0 – 4 day (1–31) 5 – 8 month (1–12) 9 – 15 year (0–99) Validity start date can be calculated from RtsExpDate & the number of days of validity indicated by RtsProductType.

Table 11—Pass and Transfer Product Objects (continued)

Field	Size	Values	Pos.	Description
RtsExpDate (continued)	16	Date () or zero	9-24	If 'RtsProductType' indicates an Open Dated [Rolling Period] Pass Product, then RtsExpDate is initially encoded with an expiry date that is further in the future than Purchase Date plus the Product Validity Period'. Open Dated Passes normally don't last indefinitely and are set to expire at a date related to the purchase date. Therefore, the actual encoded expiry date for open dated passes will normally be within one month to a year from product purchase date, depending on Product Type. The Open Dated Product's [RtsExpDate] is re-encoded, to match the Start [first use] Date and the validity period, on its invocation at a Gate or Validator.
RtsExpTime	11	0-1439	25-35	Time this product expires. Time in minutes past midnight. (24:00 hour clock)
RtsRemTrips/Rides	6	0-63	36-41	Number of remaining transit Trips or Rides for the current product. Does not include any trips associated with a renewal in advance. [NA to Transfers] – In the case of single transfers, this value should be set to 0.
RtsUseSequenceNumber	7	0-127	42-48	Identifies the product's use sequence number. This field is set to "0" upon "initialization." Upon first "load," this field would then be incremented to a value of "1." The Regional Administrator should establish a policy determining how all systems must treat this field when the maximum value of 127 is exceeded. As an example, the next value for this field could always be one (1), allowing a zero (0) value to be used to identify a purse that has never been loaded. Some regions may alternatively use zero as the next value in order to simplify calculations for the total number of loads performed on a single card.

Table 11—Pass and Transfer Product Objects (continued)

Field	Size	Values	Pos.	Description
RtsProductType	8	0-255	49-56	The Regional Administrator and the participating transit agencies within the region define the Pass Product type codes. The applicable code is posted to the PICC when it is Autoloaded or when the customer buys the Pass product at a vending machine, ticket booth, or other device. There are up to 253 possible pass product codes for each Agency & up to 253 for regional pass products. Once selected, these codes are normally fixed and permanent for the duration of the product's life. A product type code cannot be changed until all 'on card products' of that type have expired. The RtsProductType code, together with other data elements such as RtsExpDate & RtsExpTime are captured, by the fare gate CID, and used in transaction records for accounting, demographic reporting, and other downstream fare collection system processing. For example only: (New York City Transit (NYCT) Metro PICC) 0 = T-Purse or [SV] in all cases 1 = 2 Ride Pass 2 = Weekly Off peak Pass 3 = Rolling monthly unlimited pass 4 = Rolling monthly Transit Center pass 5 = 3-day unlimited tourist pass 6 – 253 = additional products 254 = AutoValue 255 = Account Linked

Table 11—Pass and Transfer Product Objects (continued)

Field	Size	Values	Pos.	Description
RtsLocationEncoding	32	0-4,294,967	57-88	Indicates the location validity of the Pass Product within the Region. The interpretation of this field is dependent on the RtsLocationEncodingType field. Examples: Sector/Route/Sector Encoding Bits Denotation 0 – 10 Valid Sector a 11 – 21 Valid Sector b 22 – 31 Valid Route Number connecting Sectors a & b Point to Point Encoding: Bits Denotation 0 – 15 Point a = RtsLocationID a [0-65,535] 16 – 31 Point b = RtsLocationID b [0-65,535] Location IDs should be assigned by each agency to each CID within its system and a Master List shall be maintained by the Regional Administrator to avoid duplication of values within a region.
RtsCIDTransactionNumber	7	0-127	89-95	This data element provides a unique number for each transaction that occurs with the CID. This data element will be the 7 least significant bits of the CID Transaction sequence number.
RtsCIDID	16	0-65,535	96-111	CID ID. Unique Identity [within the Region] of the Device's CID.
DACP&T or CRCP&T Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65,535	112-127	DAC for authentication & error detection or CRC for error detection only Calculated on all P&TPOs and P&TPOXs.
Total	**128**			

NOTE—The P&TPO is a core object required for FF PICCs. At least one Product Object shall be associated with a Product Index Object implementation.

5.11.1 Pass and Transfer Product Object User's Information

5.11.1.1 RtsExtensionStatus

This data element is used to signify that a P&TPOX is associated with this P&TPO. The single bit can represent a status of no extension if [0], or a [1] indicates an extension is present. In the event that an extension exists, the application must read all 128 bits of the extension to determine what the extension contains. An extension may contain data that is needed to complete the actual Product (e.g., a Pass product might have an extension that defines a specific route where the Pass product may be used).

5.11.1.2 RtsAutoloadSubscribed

This data element provides for two bits that represent either a [0] for no Autoload subscription or a [1] for Autoload subscribed. This bit shall be set by the application or at the time of initialization. If selected for an Autoload subscribe the Pass product will be reloaded by the Autoload function based upon the information provided in other data elements and the PIO. This data element shall be used in conjunction with the following data elements in this Object: [RtsRenewedInAdvance], [RtsExpDate] and [RtsPaymentType] so that the application has the appropriate data necessary to validate the subscribed selection.

5.11.1.3 RtsPaymentType

This data element identifies the type of funds used to purchase the original product value. Two (2) bits are assigned to this data element providing for four (4) fund or revenue types. This data element is of value to the customer service agent when attempting to refund payment to the patron. If the funds were purchased by credit card the funds should be credited back to the credit card. However, the agency or regional operator could create a policy that would simply use a [RtsPaymentType] [1] to represent that the payment was made with a credit card and that no refunds are allowed for credit card purchases. This data element would be written to at the time of product purchase.

5.11.1.4 RtsRenewedInAdvance

This data element allows for early renewal of a fare product. A single bit is used to establish the status of an early or renewed in advance fare product add value. The transit application would write to this data element when the patron vendor add-value or a reoccurring Autoload event took place. For example, if the patron purchases a 3-day or monthly Pass product and has used it to the 28th day of the month but decides to buy or renew the Pass product, the patron would want to be sure that the two or three days that remain on the existing Pass are not lost upon renewal. This data element informs the application that a second 30-day Pass product is also loaded on the PICC but is not in use until the existing 30-day Pass product reaches its expiration date. This data element works in conjunction with the data element in the Object [RtsExpDate]. A new 30-day P&TPO will be created upon renewal. This renewed in advance P&TPO would have this data element written to with a [1] indicating that it is a renewed in advance Pass product that should not be used until the duplicate existing 30-day Pass product reaches its expiration date.

5.11.1.5 RtsLocationEncodingType

This data element provides for a single bit identifier that represents the type of encoding used for this Product Object by the agency or regional operator. The data element in the Product Object [RtsLocationEncoding] is used in conjunction with this data element. Further explanation of sector/route

and Point-to-Point can be found in the [RtsLocationEncoding] Data element. The two modes are entitled Type-0 and Type-1.

5.11.1.6 RtsExpDate

This data element provides for a 16-bit field that is used to represent the expiration date of the P&TPO. This data element is also used to calculate the P&TPOs period of starting validity. For example, if a 7-day fare product is purchased with a [RtsExpDate] of 02/28/05, and the product type code [RtsProductType] data element is a 7-day Pass product, the transit application would calculate in the following manner:

[RtsExpDate] - [RtsProductType] (7 days) = Feb 28 minus 6 days (February 22).

It is possible to set this data element to a date far in the future if no actual expiry date is desired.

The actual values placed in this data element would be structured as follows:

Bits 0-4 are 25 = 32 possible numbers (i.e., day 28 is {11100})

Bits 5-8 are 24 = 16 possible numbers (i.e., month 02 is {0010})

Bits 9-15 are 27 = 128 possible numbers (i.e., year 05 is {0000101})

Or

(0000101001011100) = 02/28/05

5.11.1.7 RtsExpTime

This data element provides for an 11-bit counter to signify the number of minutes that may be used by the application to decrement from the date of expiry starting from midnight (24:00 hours) on the date recorded in the RtsExpDate field. This data element is normally used by the application to validate or cause to expire a transfer product issued from use of another fare product. Since this product object can be used for either a pass product or transfer product this data element may serve multiple purposes. If used in pass product mode, the data element could signify that last valid day and minute in time of operation. For example, a pass would be able to expire at 17:00 hours opposed to 24:00 hours if only [RtsExpDate] were used. However, the most valuable implementation of this data element is to calculate an hour-based transfer validly period. If a 2-hour valid transfer product were issued at 10:00 hours the transfer would actually be valid until the 720th minute past 24:00 hours were reached. The structure of this element is used in the following method:

Bits 24-34, 211 = 2048 possible minutes; or,

720 minutes past midnight equals 12:00 hours; or,

(01011010000) = 720 minutes past 24:00 hours

5.11.1.8 RtsRemTrips/Rides (Journeys or Boardings)

This data element provides a 6-bit field used to keep the status of the Pass products associated Trips or Rides. The 6-bits allow for a maximum of 63 trips or rides to be registered in this field per Product Object. It is common to see Pass products containing 2, 6, 10, and even 12 rides. However, it is unlikely to have a Pass Product with more than 24 rides. Therefore, the 63 possible rides should more than cover all requirements. The application would read this data element to verify whether any rides existed in association with this Pass product. It would then calculate from the 6-bit representation how many trips or rides were remaining. If rides or trips remained, depending on what type of Pass product, the application would reduce one ride or more rides and re-write this data element with the remaining trips or rides. The

application typically also uses this data element to report to the patron via display the quantity of remaining trips or rides. This data element is not used when this P&TPO is used for transfers since a transfer does not contain trips or rides.

5.11.1.9 RtsUseSequenceNumber

This data element makes a provision to track the product to a sequence number. The tracking of the Product with such a number may be employed as an additional method of preventing fraud. A break in the products sequence number shall be interpreted as a potentially lost or missing transaction.

5.11.1.10 RtsProductType

This data element provides an 8-bit field that supports a maximum of 253 products per agency or regional organization. Product type [0] is reserved for the T-Purse or Stored Value product. A Product type within the PICC is removed only when the product has expired and or permanently removed or blocked.

5.11.1.11 RtsLocationEncoding

This data element provides the encoding representation for a Stop Point or a Route. If encoded as a route and the route direction is important (i.e., Northwest) then this route could be allocated two route IDs in a fare table such as:

Route 103 NW = encoded route 2048

Route 105 SE = encoded route 1055

If this data element used as a unique stop point ID then this stop could be implemented as:

Route 103 NW Stop point 10 = encoded 20072 and so on…

5.11.1.12 RtsCIDTransactionNumber

This data element provides a unique number for each transaction that occurs with the CID. The [RtsCIDTransactionNumber] is used by the transaction acquirer or agency of use to check for missing transactions at their devices in a similar way to that which the RtsUseSeqNumber is used by a product issuer or owner to track missing transactions. This data element is used in conjunction with [RtsCIDID] to track the unique transaction number to the actual CID performing the transaction. Each transaction data packet that is completed and sent up to the back end system for accounting and settlement shall contain both [RtsCIDID] and this data element for tracking purposes. It is imperative that the CID Transaction Number is not corrupted due to power loss.

5.11.1.13 RtsCIDID

This data element provides for unique serialization of each CID within the agency or region. Sixteen bits are provided to allow for 65,535 unique IDs. Since the highest known CID count in any one region is estimated to be less than 30,000 this data element's ID count should be sufficient to cover all regional possibilities. Anytime a transaction occurs, this data element must be read and sent up to the back end system as part to the packaged data. This is especially important each and every time a CID is used to load value onto a PICC. The CID ID in joint with the actual transaction data provides the necessary information to help identify the transaction and to prevent fraud. Each CID must have a unique number assigned at the time of or before field deployment. A responsible implementation method would involve assigning a correlation between the CID ID and the actual physical location of the CIDs deployment through a lookup

table or other method. It is important to know not only which CID performed the transaction but where the CID is physically located.

5.11.2 Pass and Transfer Product Object Extensions

5.11.2.1 General

The P&TPOX can be used to store extra data pertaining to any product as described in Table 12. The P&TPOX has a one-to-one relationship with the appropriate product. This is the object mechanism used to support products such as Routes or Multiple Journeys associated with a given fare product.

FF Guidelines:

— Each P&TPOX is an extension of a P&TPO and has a one-to-one relationship with the appropriate P&TPO.

— Each P&TPOX shall be associated with one and only one P&TPO.
NOTE A Product Object DOES NOT require the use of a Product Extension Object. The existence of a Product Extension Object is dependant on the product type.

— There can be one P&TPOX per P&TPO.
NOTE PICC data memory or transaction performance or both could constrain the number of Extension Objects permitted. Each Product Extension Object shall be no greater than 128 bits (16 bytes).

— A P&TPOX will be automatically updated when the corresponding P&TPO is updated.

— A Product Extension Object may not be deleted directly. It can only be deleted when the associated Product Object is deleted (i.e., cascade deletion).

By way of example, a P&TPOX associated with a Pass Product could be a Route Object that defines the segments of the customer's route for any particular transit fare product. For a transit agency, this can define an origin via the destination segment or sector. For a regional product, the system can define the segments from which patrons have traveled. The routing contained within the Route Object will be used to provide proof-of-payment information for non-gated transit systems as well as travel demographics. This object contains multiple connecting segments or zones that define the beginning, connecting and terminating segments for a trip. This object containing the route provides up to a maximum of seven connecting segments. This routing can be utilized for transit modes that identify trips by their inter-connecting points, such as rail and airlines.

— The Region shall identify each destination station or point with a unique numeric value for regional journey.

— Each segment represents the precise valid routes that the patron will travel.

Table 12—Pass & Transfer Product Object Extension

Field	Size	Values	Pos.	Description
RtsP&TPOXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionType	6	0-63	2-27	Identifies the type of P&TPOX This field would be defined when an applicable field is determined by the Standard body. Currently, the Standard does not define any of the values. Additional values and extension types will be defined as the Standard evolves. The definition of the extension type will define the data elements within this extension.
RtsExtensionData	120		8-127	Product Extension Data
Total	**128**			

5.11.2.2 Transfers

A transfer is a valuable fare payment instrument widely used by agencies around the world. A transfer may provide a free or discounted pass or even multiple passes. This is usually accomplished when a primary fare product is initially used. For example, if a patron uses a T-Purse to gain access to a rail system the appropriate fare is deducted and the patron may be given a transfer or multiple transfers. The most efficient method of providing a transfer is to electronically write a transfer onto the PICC at the same time the patron is gaining access to the transit system. It would be convenient to display for the patron that a dollar amount was deducted from the T-Purse and one or more transfers have been granted. Upon entrance to another rail station, bus or ferry the patron is granted access for no additional charge or at a discounted charge. In most cases, transfers are time sensitive, meaning that transfers granted at a given time may only be valid for a limited period of time, such as one or two hours. Most applications grant transfers for a period of less than three hours.

Transfers may be supported using two approaches. The first approach, Associated Transfers, fully associates the transfer(s) with the primary fare product. The second, New Product Object Transfers, creates a new fare product that defines the transfer as separate from the primary fare product. Each method has its advantages and disadvantages.

5.11.2.3 Associated Transfers

This type of transfer uses less memory since it does not require creation of a separate product object or fare product. It also requires less transaction time since fewer read and write cycles are required. The Associated Transfer makes use of the Transaction History Object to maintain association between the primary fare product and the granted transfers. The disadvantage is that the method is relatively complex to comprehend and the implementer must take care in reading the appropriate Transaction History log so that the transfer is not confused or lost.

One way to look at this approach is to have a primary fare product that becomes tagged with flags that indicate that transfers are available. In Table 13 the use of two granted transfers is illustrated. Per the "Use 3" column, the use of the second transfer is interrupted by the patron's activity of purchasing a Pass Product with a value of $10.00. An interrupted transfer can create implementation problems since the primary fare product associated with the transfers held in the THO is no longer the last transaction fare product. When the patron enters the transit system again he would expect to use his last transfer since it is still within the allowed validity time. In "Use 4" the Transfer Code and Transaction Linked are used in conjunction to maintain association with the primary fare product. The example in this table uses a three-hour transfer expiration time, so the second and third transfers are granted. The second transfer is used on an express bus, and this example shows a discounted transfer fee of $0.50.

Table 13—Example – History Object Records Associated with SV Use Transfers

History Object Data Element	Use 1 Entry	Use 2 Entry	Use 3 Pass Load	Use 4 Entry	Use 5 Entry
TransactionType	6	6	1	6	6
AgencyID [of Use]	3	3	4	4	4
ProductType	0	0	0	0	0
LocationID	43	45	50	50	60
Time	8.00	9.15	10.00	10.05	10.45
TransferStartTime	8.00	8.00	8.00	8.00	8.00
TransactionLinked	0	3	2	3	3
Transfer Code	27	27	36	31	31
TransactionValue	2.00	0.00	10.00	0.50	0.00

Use 1 = SV Entry at station 43 @ 8.00
Use 2 = Boarding Bus @ 9.15 [within transfer time]
Use 3 = Purchasing a new Pass Product
Use 4 = Boarding an Express Bus [within transfer time but with 0.50 upgrade charge]
Use 5 = Entry at Station 60 @ 10.45 [within transfer time]

Table 14 shows a transfer example using an interrupted transfer sequence with the patron purchasing Stored Value or T-Purse value. Activities are similar to those shown in the above Table 13 example.

Table 14—History Object Records Associated with Pass Use Transfers

History Object Data Element	Use 1 Entry	Use 2 Entry	Use 3 Pass Load	Use 4 Entry	Use 5 Entry
TransactionType	4	4	6	4	4
AgencyID [of Use]	3	3	4	4	4
ProductType	5	5	0	5	5
LocationID	43	45	50	50	60
Time	8.00	9.15	10.00	10.05	10.45
TransferStartTime	8.00	8.00	8.00	8.00	8.00
TransactionLinked	0	3	2	3	3
Transfer Code	27	27	36	31	31
TransactionValue	0.00	0.00	5.00	0.50	0.00

Use 1 = Entry at station 43 @ 8.00
Use 2 = Boarding Bus @ 9.15 [within transfer time]
Use 3 = Purchasing Purse Value
Use 4 = Boarding an Express Bus [within transfer time but with 0.50 upgrade charge]
Use 5 = Entry at Station 60 @ 10.45 [within transfer time]

5.11.2.4 New Product Object Transfers

This method creates a new fare product (Transfer) when a primary fare product is used and the fare table policy grants these transfers. When this occurs, the transit application creates a new fare product that is classified as a transfer. Upon patron entrance to the transit system the system looks for a valid transfer product.

5.11.2.5 New P&TPO Supporting Interrupted Transfers

A new P&TPO may be created and used in accordance with the following scenario:

An initial boarding or ride is undertaken by the patron, and a Transfer 1 is created in the form of a new product in accordance with applicable transfer rules. The Transfer 1 is then used for a single free ride or boarding in sub-region A. Such transfer may be free or discounted (the latter requiring an additional payment) or a multiple (e.g., three) ride transfer within a designated time period (a multiple ride transfer would result in deductions of remaining trips). If the patron enters sub-region B, he may be unable to use transfer 1, which may not be valid outside sub-region A. The patron must then pay for the ride using an applicable product (e.g., T-Purse or Account Linked). A Transfer 2 may be created at this point based on applicable rules.

This Transfer 2 may be updated by writing to a separate fare product or product slot, or it may update Transfer 1's original Product Object to form a "Transfer 1 & 2" (with perhaps limited validity) depending on the rules of the region. At a later time, the patron uses transfer 2 for a "free" (or perhaps reduced) boarding in sub-region B. In this case, the transfer is updated as required. At a time even later, the patron returns to sub-region A. If the transfer is still valid, the appropriate action is taken (i.e., ride or boarding deducted, transfer updated, etc.).

The requirements (rules) of the transfer product are similar to those of typical pass products:

A free "slot" must be available for each transfer product desired. It is preferred that transfers be limited to a single block size, although, like pass products, they may have the "point-to-point" extension. In the case of "multiple" transfers, multiple slots may be required. Alternatively, it is possible to combine transfers as mentioned above, though this is subject to the rules of the agency or region.

5.11.3 Routes and Sector Object Extensions

Routes within the P&TPOX define the segments of the patron's route for any particular transit fare product. For a transit agency, they can define an origin location via the destination segment or sector. For a regional product, the system can define the segments from which patrons have traveled. This extension of the P&TPO has a one-to-one relationship with the appropriate product. The routing contained within the extension will be used to provide proof-of-payment information for non-gated transit systems as well as travel demographics. This object contains multiple connecting segments or sectors or zones that define the starting point, connecting and terminating segments for a trip. This extension provides up to a maximum of seven connecting segments within a route. This routing can be utilized for transit modes that identify trips by their inter-connecting points, such as rail and airlines.

Table 15 demonstrates a P&TPOX configured to add routing for multiple segment journeys. A P&TPOX can be configured in various manners to fulfill the fare product application requirements. Another sample configuration is given in Table 16, which provides for sector, route to sector fare product support.

Table 15—Point-to-Point-to-Point Route Object - Example 1

Field	Size	Values	Pos.	Description
RtsP&TPOXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionType	6	0-63	2-7	Identifies the type of Product Extension.

Table 15—Point-to-Point-to-Point Route Object - Example 1 (continued)

Field	Size	Values	Pos.	Description
RtsAgencySpecial	8	0-255	8-15	Special Agency ID for this combination if required. 0 = Same as Agency indicated by Product Object.
RtsLocation1	16	0-65535	16-31	Location 1. Represents the start point for first segment of a multi-segment trip route, e.g., origin = 001 for Penn Station, New York.
RtsLocation2	16	0-65535	32-47	Location 2. Represents start point for the second segment of a multi-segment trip route, e.g., via = 100 for Newark.
RtsLocation3	16	0-65535	48-63	Location 3. 0 = not applicable.
RtsLocation4	16	0-65535	64-79	Location 4. 0 = not applicable.
RtsLocation5	16	0-65535	80-95	Location 5. 0 = not applicable.
RtsLocation6	16	0-65535	96-111	Location 6. 0 = not applicable.
RtsLocation7	16	0-65535	112-127	Location 7. Represents the end point of a multi-segment trip route, e.g., destination = 101 for Newark International Airport.
Total	**128**			

Table 16—Sector, Route, Sector Object -Example 2

Field	Size	Values	Pos.	Description
RtsP&TPOXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).

Table 16—Sector, Route, Sector Object -Example 2 (continued)

Field	Size	Values	Pos.	Description
RtsExtensionType	6	0-63	2-7	Identifies the type of P&TPOX.
RtsRoutingType	0-15	8-11	8-11	Identifies the type of Routing specified: 0 = Sector, Route, Sector [this example]. 1 = Point to Point to Point. 2-15 = Reserved for future use.
RtsAgencySpecial	8	0-255	12-19	Special Agency ID for this combination. 0 = Same as Agency indicated by Product Object.
RFUP&TPOX20-21	2	0	20-21	Reserved for future use.
RtsSector1	11	0-2047	22-32	Sector 1.
RtsRouteA	10	0-1023	34-43	RouteA between Sectors 1 & 2.
RtsSector2	11	0-2047	44-54	Sector 2
RtsRouteB	10	0-1023	55-64	RouteB between Sectors 2 & 3.
RtsSector3	11	0-2047	65-75	Sector 3.
RtsRouteC	10	0-1023	76-85	RouteC between Sectors 3 & 4.
RtsSector4	11	0-2047	86-96	Sector 4.
RtsRouteD	10	0-1023	97-116	RouteD between Sectors 4 & 5.
RtsSector5	11	0-2047	117-127	Sector 5.
Total	**128**			

5.12 Stored Value and T-Purse Product Objects

Stored Valued Products are agency-specific. The T-Purse is the regional stored value fare product. Each PICC may contain a T-Purse that can store value in local currency (default: US Dollars). This currency-like value will be acceptable at all participating transit facilities within the region. The history objects track the most recent agency stored value & T-Purse transactions. When the PICC is initialized, the Stored Value Products & T-Purse will be established with zero (0.00) balance unless the issuer elected to pre-encode value. In the case of a zero initial value, the patron must add value at time of purchase or later at an add value machine or at appropriate reload outlets. As with the transit pass products described in the previous section, the patron can subscribe to the Autoload feature for the Stored Value products and T-Purse. Figure 9 refers to the high-level structure relationship between T-Purse and SV Product Objects.

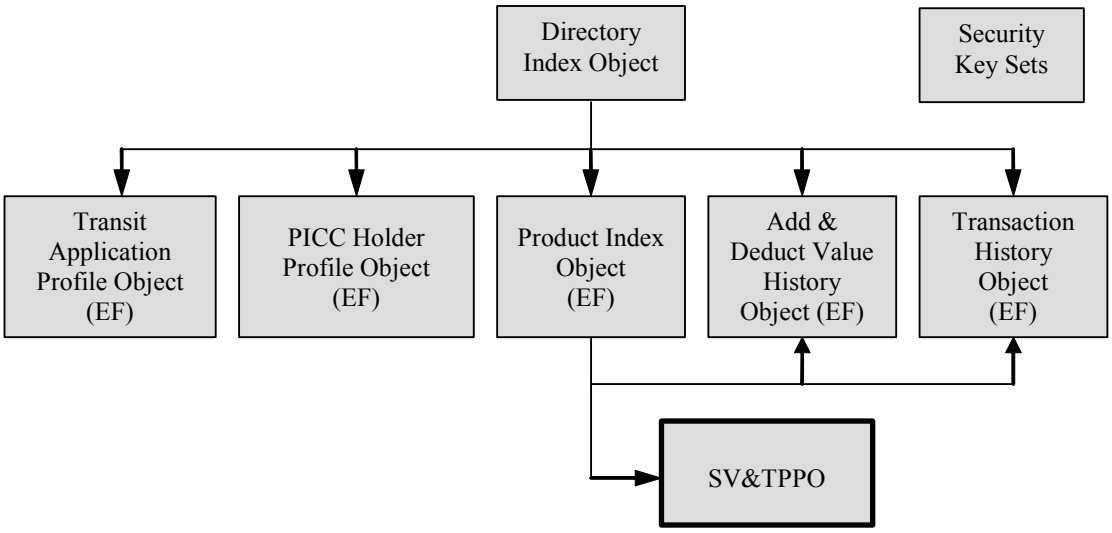


Figure 9—Stored Value and T-Purse Product Objects

FF Guidelines:

— There can only be one instance of a T-Purse per PICC.

— If the customer enables Autoload subscription services, the T-Purse product will be automatically reloaded at a CID/PCD located on fare gate, vending machine, validator or other non-transit outlet device.

— The T-Purse shall have the use of a single dedicated Key Set however; the read key may be shared with other files or objects. The combined read and write or separate write key(s) should be dedicated.

— Stored Value and T-Purse Product Objects may coexist on a Full Featured PICC but system implementation should take caution to prevent patron confusion.

— Stored Value products can only be used by a single agency. Agency-to-agency or regional value product use must utilize the T-Purse.

Table 17 represents a consolidated value based Product Object that can be configured as a SV&TPPO. All required information and data representing either a Stored Value or T-Purse product is provided in this object.

Table 17—Stored Value (SV) & T- Purse Product Objects

Field	Size	Values	Pos.	Description
RtsSV&TPPOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionStatus	1	0-1	2	0 = SV or T-Purse Object with no Extension. 1 = SV or T-Purse Object with Extension. NOTE The extension's position follows the core SV&T-Purse object.
RtsAutoSubscribe	2	0-3	3-4	This field represents the Autoload subscription indicator for this load: 0 = Field not used 1 = Threshold 2 = Recurring 3 = Recurring & Threshold
RtsValueExpires	1	0-1	5	Expiry indicator. Indicates that this is a Single Load SV product that expires on date indicated by RtsExpRecurDate. 0 = Normal 1 = Expires If a value of "1" is set for this field, it will not be possible to allow non-calendar based recurring autoloads.
RtsRemValueSign	1	0-1	6	Value designates a positive or negative balance. 0 = Positive 1 = Negative
RtsRemValue	16	0-65535	7-22	Remaining currency value.

Table 17—Stored Value (SV) & T- Purse Product Objects (continued)

Field	Size	Values	Pos.	Description
RtsExpirationRecurringDate	16	Date ()	23-38	Date that this product expires or Last Recurring Load Date. Format ddmmyy 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) 0 = No expiration If RtsValueExpires = "0," AND RtsRecurringAutoloadType is not set to "0," then this field indicates the last date when a recurring autoload was performed. If RtsValueExpires = "0," and RtsRecurringAutoloadType is set to "0," this field should be set to 0. If RtsValueExpires = "1," this field represents the expiration date of the stored value or T Purse Product. No non-calendar based recurring autoloads are allowed in this case. If RtsValueExpires = "1," this field CANNOT be set to "0."
RtsRecurringAutoloadType	3	0-7	39-41	Used to differentiate from SV Recurring Autoload types: 0 = Field is not used 1 = Weekly 2 = Monthly 3 = Bi-Annual [every 6 months] 4 = Annual 5-7 Reserved for future use Recurring autoloads are performed as defined by Regional Policy. (e.g., a region's policy could be to perform weekly autoloads on Tuesdays).

Table 17—Stored Value (SV) & T- Purse Product Objects (continued)

Field	Size	Values	Pos.	Description
RtsAutoloadThreshold	3	0-7	42-44	The value in this field is used to identify the general parameter that triggers an autoload function. 0 = Field is not used 1 = balance is equal to or less than zero down to but not less than the PICC deposit value (default, permanent standard) 2 = [Lookup Tag-1] 3 = [Lookup Tag-2] 4 = [Lookup Tag-3] 5 = [Lookup Tag-4] 6 = [Lookup Tag-5] 7 = reserved for future use NOTE Lookup tags point to configuration files, or database tables that indicate the actual value at which to perform Threshold Autoloads.
RtsSVThresholdLoadAmount	15	0-32767	45-59	Value to Add for a Threshold Autoload.
RtsSVRecurringLoadAmount	15	0-3276	60-74	Value to add for a Recurring Autoload.
RtsCurrencyCode	3	0-7	75-77	Currency of the value of this Product. The currency code is considered fixed and permanent where indicated and consistent for all regions that recognize and adhere to the transit smart PICC. The fare collection system conforming to these specifications will recognize the defined currency and deduct the equivalent of that currency from the T-purse. Codes are: 0 = Field not used 1 = US Dollar (default) 2 = Canadian Dollar 3 = Mexican Peso 4 = Pound Sterling 5 = Japanese Yen 6 = Euro 7 = Reserved for future use The currency code is defined by the RtsCountryID field, unless there is a non-zero value populated in this field.

Table 17—Stored Value (SV) & T- Purse Product Objects (continued)

Field	Size	Values	Pos.	Description
RFUSV&TPPO78-81	4	0	78-81	Reserved for future use.
RtsSVTransSeqNumber	7	0-127	82-88	Purse transaction sequence number. This field is set to "0" upon "initialization." Upon first "load," this field would then be incremented to a value of "1." The Regional Administrator should establish a policy determining how all systems must treat this field when the maximum value of 127 is exceeded. As an example, the next value for this field could always be one (1), allowing a zero (0) value to be used to identify a purse that has never been loaded. Some regions may alternatively use zero as the next value in order to simplify calculations for the total number of loads performed on a single card.
RtsCIDTransactionNumber	7	0-127	89-95	This data element provides a unique number for each transaction that occurs with the CID. This data element will be the 7 least significant bits of the CID Transaction sequence number.
RtsCIDID	16	0-65535	96-111	Issuing Machine CID ID. Used to identify the Encoding equipment.
DACSV&TPPO or CRCSV&TPPO Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65535	112-127	DAC for authentication & error detection or CRC for error detection only. Calculated on all SV&TPPO Objects including extension
Total	**128**			

NOTE—The SV&TPPO is a core object implemented on a FF PICC platform.

5.12.1 Stored Value and T-Purse Product Object User's Information

5.12.1.1 RtsExtensionStatus

If this data element is enabled [1] it allows for the SV&TPPO to carry an extension object (SV&TPPOX) containing 128 bits of additional information. One use may be for a "supportive information product that uses the T-Purse object to setup limited data elements providing the necessary information to allow the Extension Object to function in a certain capacity. If no SV&TPPOX is selected, the core SV&TPPO will use only the first 128 bits of information. (It is important to note that an SV&TPPOX will add to the overall transaction execution time since an additional 128 bits of information must be read.)

5.12.1.2 RtsAutoSubscribe

This data element enables the three modes of Autoload subscription requested either by the patron or organization issuing the product value. Threshold [1] is selected when a fare product such as Stored Value or T-Purse reaches or becomes less than a predetermined value that is indicated by the PICC value in the Object [RtsRemValue]. The comparison of this value to that of the value set in the end-equipment (such as faregate or farebox) determines whether the product is to be replenished to the predetermined amount. The data element [RtsValueSign] will also need to be taken into account in case the amount has actually gone negative.

If Recurring [2] is enabled, then the Stored Value or T-Purse will be replenished at a given point in time. For example, if $20.00 is to be reloaded onto the PICC every 15th of the month, this will continue to occur as long as the bit is enabled or a maximum PICC value stored is achieved.

The selection of Recurring and Threshold [3] simply allows for both the recurring and threshold to take place. In this case, recurring with a predetermined amount, such as that offered by a "Smart Benefits" program would be loaded on a given date. To supplement this value a threshold value may also be set up that ensures that the patron will not run out of value even if the recurring Smart Benefits fund were exhausted.

5.12.1.3 RtsValueExpires

This data element allows for an expiration date to be set for the value added to a stored value product. The SV&TPPOX makes use of the data element [RtsExpRecurDate] to determine when the value expires. If a [0] bit is set, the value within the stored value product does not expire. This field should not be used if the Regional Administrator allows the addition of value to the stored value or T-purse product after an initial load of value is performed. If the Agency or Region allows non-calendar based, recurring autoloads (autoloads that are triggered based on number of days or months since the first autoload was performed rather than on a set calendar day or date each week, month, etc.), this field must be set to 0.

5.12.1.4 RtsRemValueSign

This data element serves to inform the application that the Stored Value or T-Purse Product has entered into a state of negative value. This may occur when an agency or regional operator allows through policy the patron to take one remaining ride even though the value on the product is insufficient to cover the required fare. This is often used with a fare-by-distance system upon exiting. In the event this data element is set to [1] the negative value will be taken into account once the product is re-loaded with new value, e.g., if the product has a value of minus $1.50 and $20.00 is reloaded, the actual positive value for this product will be $18.50. In addition, the [1] also indicates to the fare logic that the product is already considered negative and entry into the system is not allowed until the product is set to a positive value. The bit must be used in conjunction with fare value usage calculations to ensure whether the value of the product is positive or negative. The data element [RtsRemValue] is used in conjunction with this data element.

5.12.1.5 RtsRemValue

This data element indicates the Stored Value or T-Purse product remaining value. This data element is written or updated every time the fare product is used. This data element may be represented as a unit value such as: one-cent, one nickel, one-dime, one-quarter or one-dollar. The value in this data element is used to inform the patron of remaining value and to track the product's value. Since this data element is simply a value, it must be used in conjunction with [RtsRemValueSign] so that a positive or negative value can be indicated.

5.12.1.6 RtsExpirationRecurringDate

This data element sets a date indicating when the last Autoload occurred or when the value in a stored value or T-purse product expires. The 16 bits in this data element are used to establish this date by assigning representative digits. Since the year field uses only 2 digits, care needs to be taken in representing only dates that are of the year 2000 to 2099 where "00" is the year 2000. In the event all zeros are placed in the digits, this will indicate that there is no date set for an Autoload reoccurring event. The 16-bits are implemented as follows:

Superscript

Bits 0-4 are 25 = 32 possible numbers (i.e., day 28 is {11100})

Bits 5-8 are 24 = 16 possible numbers (i.e., month 02 is {0010})

Bits 9-15 are 27 = 128 possible numbers (i.e., year 05 is {0000101})

Or

(0000101001011100) = 02/28/05

As a note, it is not possible to support both the expiration of value stored within a stored value or T-Purse product AND a non-calendar based recurring autoload. Agencies and implementers should be aware of this requirement. However, in most cases, it is unlikely that the value within a Stored Value or T Purse product will expire on a full-featured PICC since reloading of value to those products would be desirable.

5.12.1.7 RtsRecurringAutoloadType

This data element is used as a high-level indicator to differentiate the Stored Value or T-Purse Autoload classification. Three bits are used to indicate that the reoccurring Autoload event is of type Weekly or Monthly. The reserved values may be used to indicate other Autoload event types. This data element may be used in conjunction with [RtsExpRecurDate] to indicate, as an example, an Autoload Stored Value or T-purse product being reloaded on a monthly basis but only on a specified date.

5.12.1.8 RtsAutoloadThreshold

This data element is normally reserved for use with the T-Purse but may be used with a Stored Value product. Three bits are assigned providing eight different indicators representing when a Threshold Autoload should occur, e.g., when set to [2] the application is informed that the T-Purse product should be reloaded since the threshold is now less than $5.00 in value. The application will reload the product with the value set in the [RtsSVThreshholdLoadAmount] data element. This reloaded value is considered a post authorized transaction since the value is reloaded before the actual transaction is approved. This feature provides the patron with an uninterrupted fare product usage as long as the financial instrument attached to the Autoload event is valid.

5.12.1.9 RtsSVThreshholdLoadAmount

This data element provides for the actual value to be reloaded when an Autoload threshold event occurs. This data element provides for reload amounts to be set for individual thresholds since this value is maintained on each patron's PICC. Fifteen bits are allocated to provide for the actual unit currency value applied such as: one-cent, one nickel, one-dime, one-quarter or one-dollar, where US dollars are selected. When the data element [RtsAutoloadThreshold] is used and a set threshold is met, the application will read this data element and proceed to reload the SV or T-Purse product with the value set.

5.12.1.10 RtsSVRecurringLoadAmount

This data element provides for the actual value to be reloaded when an Autoload reoccurring event is triggered. This data element provides for individual reload amounts to be set as reoccurring since this value is maintained on each patron's PICC. Fifteen bits are allocated to provide for the actual unit currency value applied such as: one-cent, one nickel, one-dime, one-quarter or one-dollar where US dollars are selected. When the data element [RtsRecurringAutoloadType] is used and a set reoccurring is met the application will read this data element and proceed to reload the SV or T-Purse product with the reoccurring value set. The only difference between a Threshold and Reoccurring value set is that a reoccurring is time based opposed to value based for the actual Autoload reload event.

5.12.1.11 RtsCurrencyCode

This data element provides for the application indication of currency type being applied to the SV or T-Purse product. Three bits are allocated offering selection of 8 different currencies. Value [0] is reserved so that no currency is unintentionally defaulted in the event a currency was not selected. This currency code must be used and the appropriate currency selected to validate the use of other data elements such as [RtsSVThreshholdLoadAmount], [RtsSVRecuringLoadAmount], [RtsAutoloadThreshold], and [RtsRemValue].

5.12.1.12 RtsCIDTransactionNumber

This data element provides a unique number for each transaction that occurs with the CID. The [RtsCIDTransactionNumber] is used by the transaction acquirer or agency of use to check for missing transactions at their devices in a similar way to that which the RtsSVTransSeqNumber is used by a product issuer or owner to track missing transactions. This data element is used in conjunction with [RtsCIDID] to track the unique transaction number to the actual CID performing the transaction. Each transaction data packet that is completed and sent up to the back end system for accounting and settlement shall contain both [RtsCIDID] and this data element for tracking purposes. It is imperative that the CID Transaction Number is not corrupted due to power loss.

5.12.1.13 RtsCIDID

This data element provides for unique serialization of each CID within the agency or region. Sixteen bits are provided to allow for 65,535 unique ID's. Since the highest known CID count in any one region is estimated to be less than 30,000 this data element's ID count should be sufficient to cover all regional possibilities. Any time a transaction occurs, this data element must be read and sent up to the back end system as part to the packaged data. This is especially important each and every time a CID is used to load value onto a PICC. The CID ID in conjunction with the actual transaction data provides the necessary information to help identify the transaction and to prevent fraud. Each CID must have a unique number assigned prior to or at the time of field deployment. A responsible implementation method would involve assigning a correlation between the CID ID and the actual physical location of the CIDs deployment through a lookup table or other method. It is important to know not only which CID performed the transaction but where the CID is physically located.

5.12.2 Stored Value and T-Purse Product Object Extension User's Information

Table 18 describes the elements in the stored value product object extension (SVPOX). (See Stored Value and T-Purse Object User's Information)

Table 18—Stored Value Product Object Extension (SVPOX)

Field	Size	Values	Pos.	Description
RtsSVPOXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsSVExtensionType	6	0-63	2-7	Identifies the type of Stored Value Product Extension.
RtsExtensionData	120		8-127	Stored Value Product Extension Data.
Total	**128**			

NOTE For SVPOX user's information please see SVTO user's information in Section 5.12.1.

5.13 Account Linked Product Object

The Account Linked (AL) product (see Figure 10) is designed to act much like a T-Purse product, except that it will not require pre-funding of the object. When the AL product is used, the CID will calculate the fare under the rules for the T-Purse but will not deduct any value from the RtsRemValue field. The DIO will indicate whether there is an ALPO in this particular transit application. Care must be taken by the implementer of the system to determine the logical "placement" of the ALPO. Possible implementations may include reserving specific product slots for ALPO and T-Purse.

Since AL follows the rules of the T-Purse product, it supports all forms of discounts based upon the transfer rules, rider class, time of day, day of week, or other concessions supported by the Standard.

AL may require a Reference Object containing the Bankcard Primary Account Number and bankcard expiration date, which are used to associate all use of this product with the corresponding bankcard account.

The AL objects also include data elements, which support implementation of a unique feature of on-card velocity checks. Since fare payment in an AL implementation is received after the patron has received or

used transit services, typical velocity protections can only be applied to the transactions when the back end system batches and provides the data. The on-card or local CID velocity check feature allow the PICC and CID to perform offline velocity checks at the time of the transaction to protect the cardholder and the service operators against unauthorized use of lost or stolen cards.

Note that AL can replace the T-Purse product function in any fare payments program however it is permitted to have a T-Purse and an AL product coexisting on the card.

FF Guidelines:

— There can only be one instance of AL per PICC.

— A patron may enable AL at a customer service facility under a controlled environment.

— AL, T-Purse, and Agency Stored Value Product Objects may coexist on a Full Featured PICC but system implementation should take caution to prevent patron confusion.

— AL is a regional product.

Table 19 represents the data elements that support an ALPO. The AL Reference Object (ALRO) is described in Section 5.14.

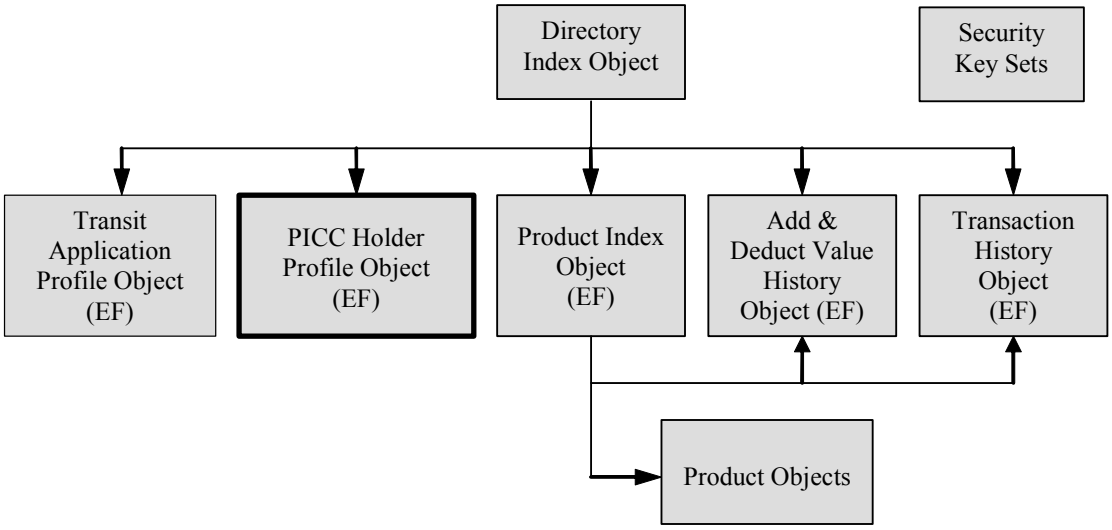


Figure 10—Account Linked Product Object

Table 19—Account Linked Product Object (ALPO)

Field	Size	Values	Pos.	Description
RtsALPOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RFUALPO2-7	6	0	2-7	
RtsDayUsedValueLimit	16	0-63000 In 1 Unit increments (units)	8-23	Accumulated value used in the current day. This value is used to perform value based velocity checks and is reset to zero on the first use in each new day. If the value within this field exceeds the RtsConsumerValueLimit or a system-wide value specified by the CID host device, the transaction is denied.
RtsCurrentDayDate	16	Date ()	24-39	Date that RtsDayUsedValue was last reset to zero. This field is updated with the current date when it is not equal to current date. Format ddmmyy 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) 0 = Not Applicable NOTE When this product is being used, this field is compared to the current date and if not equal, the RtsDayUsedValue must also be set to zero and this field is set to the current date.

Table 19—Account Linked Product Object (ALPO) (continued)

Field	Size	Values	Pos.	Description
RtsCountTimePeriod	11	0-1439	40-50	The region may also set a time period that defines a period shorter than a day for which a transaction limit may be specified. This field defines length of that time period in minutes and is used in conjunction with RtsCountTimePeriodLimit (see ALRO, Section 5.14). 0 = Field not used
RtsPeriodStartTime	11	0-2359	51-61	This field can either be used to represent the time the period starts within the RtsCurentDayDate or a time when the AL product was first used within the RtsCurrentDayDate. The regional administrator needs to specify which of these two options this field is going to represent.
RFUALPO62-63	2		62-63	Reserved for future use.
RtsPeriodCount	5	0-31	64-68	This is the count of transactions performed on this product since the RtsPeriodStartTime was set. If this count exceeds the RtsPeriodCountLimit or a value specified by the CID host device, the transaction is denied. A new period count is started when RtsPeriodStartTime is reset.
RtsCurrencyCode	3	0-7	69-71	Currency of the value of this Product. The currency code is considered fixed and permanent where indicated and consistent for all regions that recognize and adhere to the transit smart PICC. The fare collection system conforming to these specifications will recognize the defined currency and deduct the equivalent of that currency from the T-purse. Codes are: 0 = Field not used 1 = US Dollar (default) 2 = Canadian Dollar 3 = Mexican Peso 4 = Pound Sterling 5 = Japanese Yen 6 = Euro 7 = Reserved for future use

Table 19—Account Linked Product Object (ALPO) (continued)

Field	Size	Values	Pos.	Description
RtsCurrencyCode (continued)	3	0-7	69-71	The currency code is defined by the RtsCountryID field, unless there is a non-zero value populated in this field.
RtsALPOSequenceNumber	7	0-127	72-78	ALPO transaction sequence number. This field is set to "0" upon "initialization." Upon first "load," this field would then be incremented to a value of "1." The Regional Administrator should establish a policy determining how all systems must treat this field when the maximum value of 127 is exceeded. As an example, the next value for this field could always be one (1), allowing a zero (0) value to be used to identify a product that has never been loaded. Some regions may alternatively use zero as the next value in order to simplify calculations for the total number of transactions performed on a single product
RFUALPO79-88	10	0	79-88	Reserved for Future use.
RtsCIDTransactionNumber	7	0-127	89-95	This data element provides a unique number for each transaction that occurs with the CID. This data element will be the 7 least significant bits of the CID Transaction sequence number.
RtsCIDID	16	0-65535	96-111	Machine CID ID. Used to identify the Encoding equipment. The ID must be unique for CID's within a region.
DACALPO or CRCALPO Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65535	112-127	DAC for authentication & error detection or CRC for error detection only.
Total	**128**			

NOTE—The ALPO is a core object required for Full-Featured PICCs.

5.13.1 ALPO User's Information

5.13.1.1 RtsDayUsedValue

This data element provides an accumulating value that may be evaluated against RtsConsumerValueLimit in ALRO or, if specified, a region-wide value within the CID host device to ensure that use of the AL product does not exceed the regionally defined maximum. RtsConsumerValueLimit can be predetermined for each account. If the accumulated value exceeds the lower value of RtsConsumerValueLimit or a region-wide value within the CID host device, then product use is denied as a means of providing velocity checking to minimize risk.

5.13.1.2 RtsCurrentDayDate

This data element provides the field necessary to set the current date thus providing a date of reference. This data element must be set to the correct date upon first use each day to reflect the CIDs or hosting processor's date or the date of both. If zero (0) is selected, the data element is not used. Therefore, neither are the RtsDayUsedValue or RtsPeriodCount. Resetting this data element will require that RtsPeriodStartTime, RtsPeriodCount and RtsDayUsedValue also be reset.

5.13.1.3 RtsCountTimePeriod

This data element provides for a period of less than one day, enabling velocity checks within a set time frame. E.g., the period may be set to monitor excessive transaction activity during rush-hour periods. The count time period may be set to zero (0) if not used. This data element works in conjunction with RtsPeriodStartTime and RtsPeriodCount.

5.13.1.4 RtsPeriodStartTime

This data element provides for a start time that signifies the start of a period within the set day. This start time determines when RtsCountTimePeriod begins.

5.13.1.5 RtsPeriodCount

This data element provides for a count of the transactions that occurred during the period. The RtsPeriodCountLimit data element located in the ALRO or a region-wide value specified by the CID host device is used, in conjunction with this field, to limit the count within a given period. If not used this data element and RtsCountTimePeriod must be set to zero (0).

5.13.1.6 RtsCIDTransactionNumber

This data element provides a unique number for each transaction that occurs with the CID. The [RtsCIDTransactionNumber] is used by the transaction acquirer or agency of use to check for missing transactions at their devices in a similar way to that which the RtsSVTransSeqNumber is used by a product issuer or owner to track missing transactions. This data element is used in conjunction with [RtsCIDID] to track the unique transaction number to the actual CID performing the transaction. Each transaction data packet that is completed and sent up to the back end system for accounting and settlement shall contain both [RtsCIDID] and this data element for tracking purposes. It is imperative that the CID Transaction Number is not corrupted due to power loss.

5.13.1.7 RtsCIDID

This data element provides for unique serialization of each CID within the agency or region. Sixteen bits are provided to allow for 65,535 unique ID's. Since the highest known CID count in any one region is estimated to be less than 30,000 this data element's ID count should be sufficient to cover all regional possibilities. Anytime a transaction occurs, this data element must be read and sent up to the back end system as part to the packaged data. This is especially important each and every time a CID is used to load value onto a PICC. The CID ID in joint with the actual transaction data provides the necessary information to help identify the transaction and to prevent fraud. Each CID must have a unique number assigned at the time of or before field deployment. A responsible implementation method would involve assigning a correlation between the CID ID and the actual physical location of the CIDs deployment through a lookup table or other method. It is important to know not only which CID performed the transaction but where the CID is physically located.

5.14 Account Linked Reference Object

The ALRO, described in Table 20, is an object containing the AL reference information that requires secure access.

Table 20—Account Linked Reference Object (ALRO)

Field	Size	Values	Pos.	Description
RtsALROVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsConsumerValueLimit	6	0-63 in 10 Unit increments (units)	2-7	This is the daily limit of AL fare payments that can be performed for the PICC. The Region may also implement a global value velocity check at the CID host device that overrides this PICC specified limit, if the global value is lower. The maximum dollar value for this field will vary depending on the value applied by the system for one unit. As an example, the maximum is $630 if each unit is equal to $10. 0 = No limit

Table 20—Account Linked Reference Object (ALRO) (continued)

Field	Size	Values	Pos.	Description
RtsBankcardPAN	64	0 – 999 9999 9999 9999 9999	8-71	This field contains the Bankcard (credit or debit card) Primary Account Number, which, will be used by the Back end system to generate the settlement file or bankcard payment request message.
RtsBankcardExpDate	11	0100-1299	72-82	Bankcard expiration date used by the system to generate the settlement file or bankcard payment request message. Format mmyy 72-75 = month (1-12) 76-82 = year (0-99) 0 = No expiration
RtsPeriodCountLimit	5	0-31	83-87	This field specifies the transaction count limit for the RtsPeriodCount for this PICC. The Region may also implement a region-wide period count limit at the CID host device that would override this PICC element 0 = No limit
RFUALRO88	1	0	88	
RtsCIDTransactionNumber	7	0-127	89-95	This data element provides a unique number for each transaction that occurs with the CID. This data element will be the 7 least significant bits of the CID Transaction sequence number.
RtsCIDID	16	0-65535	96-111	Issuing Machine CID ID. Used to identify the Encoding equipment.
CRC/DAC	16	0-65535	112-127	DAC for authentication & error detection Or CRC for error detection only.
Total	**128**			

5.15 ALPO Extension

The ALPO Extension (ALPOX) Object, described in Table 21 can be used to store extra data pertaining to the ALPO. The ALPOX has a one-to-one relationship with the ALPO.

FF Guidelines:

— The ALPOX is an extension of an ALPO.

— The ALPOX shall be no greater than 128 bits (16 bytes).

— The ALPOX will be automatically updated when the corresponding ALPO is updated.

— The ALPOX may not be deleted directly. It can only be deleted when the associated ALPO is deleted.

User Information is not provided for either ALROX or ALPOX.

Table 21 —ALPO Extension Object

Field	Size	Values	Pos.	Description
RtsALPOXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionType	6	0-63	2-7	Identifies the type of ALPOX. This field would be defined when an applicable field is determined by the Standard body. Currently, the Standard does not define any of the values. Additional values and extension types will be defined as the Standard evolves. The definition of the extension type will define the data elements within this extension.
RFUS8-127	120		8-127	Reserved for future use.
Total	**128**			

5.16 ALRO Extension

The ALRO Extension (ALROX) Object, described in Table 22, can be used to store extra data pertaining to the AL product. The ALROX has a one-to-one relationship with the ALRO.

FF Guidelines:

— The ALROX is an extension of an ALRO.

— The ALROX shall be no greater than 128 bits (16 bytes).

— The ALROX will be automatically updated when the corresponding ALPO is updated.

— The ALROX may not be deleted directly. It can only be deleted when the associated ALPO is deleted.

Table 22—ALRO Extension Object

Field	Size	Values	Pos.	Description
RtsALROXVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).
RtsExtensionType	6	0-63	2-7	Identifies the type of ALROX. This field would be defined when an applicable field is determined by the Standard body. Currently, the Standard does not define any of the values. Additional values and extension types will be defined as the Standard evolves. The definition of the extension type will define the data elements within this extension.
RFUS8-127	120		8-127	Reserved for future use.
Total	**128**			

5.17 AutoValue Product Object

Table 11 and Table 23 describe the core and extension objects respectively to support AutoValue products. This product object is implemented when an agency or regional operator wants to provide incentives for frequent use or bulk fare purchases.

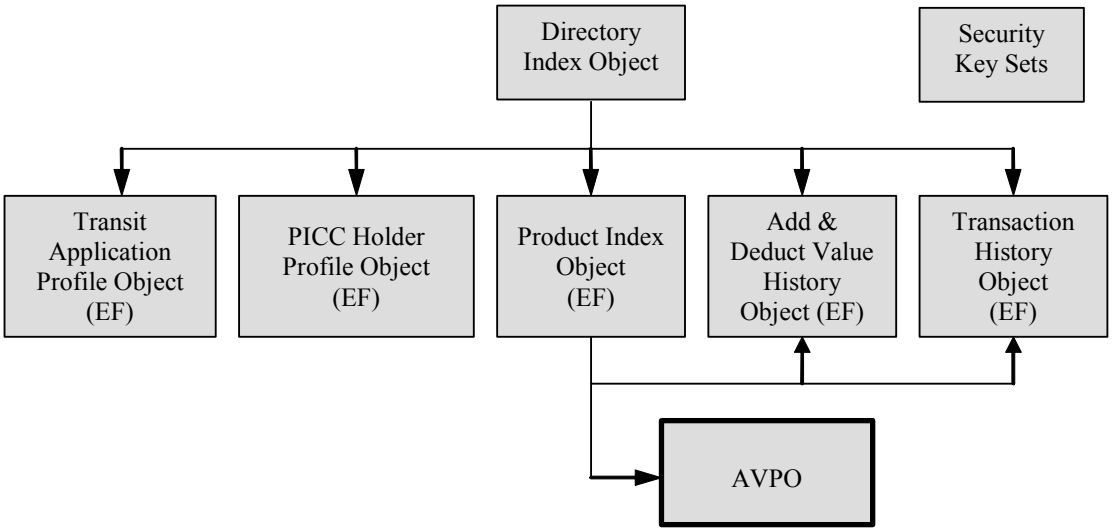


Figure 11—Auto Value Product Objects

Table 23—AutoValue Product Object (AVPO)

Field	Size	Values	Pos.	Description
RtsAVPOVersionID	2	0-3	0-1	This data element records a version number for the data object. Use of this data element enables a region to change the content (bit locations, data elements, valid values, etc.) for the data object for a particular group of cards or for a new generation of cards while allowing all previous generations or groups of cards (up to a total of 4 unique groups or generations) to continue to be used. Use of this data element requires each participating system to recognize and accommodate the differences in data structure for the object. If not used, the value for this data element shall be 0 (zero).

Table 23—AutoValue Product Object (AVPO) (continued)

Field	Size	Values	Pos.	Description
RtsExtensionStatus	1	1	2	0 = No AVPO Extension (AVPOX) 1 = AVPOX required NOTE the extension position must follow the core AVPO.
RtsAutoValueType	5	0-31	3-7	Type of AutoValue Product is represented by this data element. This version would be indicated by a [1]: 0 = Reserved 1 = Valued Fare in USD 2 = User defined 3-24 = User defined 25-31 = Reserved
RtsAgencyID	8	0-255	8-15	Agency Identity within this Region.
RtsDate	16	Date () or 0	16-31	Format date: ddmmyy 0-4 = day (1-31) 5-8 = month (1-12) 9-15 = year (0-99) Default value if field is unused is "0"
RtsUnitsToday	16	$0-655.35 in $.01 or $0-3,276.75 in $.05	32-47	Accumulation of Value Units Today.
RtsUnitsWeek	16	$0-655.35 in $.01 or $0-3,276.75 in $.05	48-63	Accumulation of Value Units 'This Week'.
RtsUnitsWeek	16	$0-655.35 in $.01 or $0-3,276.75 in $.05	64-79	Accumulation of Value Units 'This Week – 1'.
RtsUnitsWee	16	$0-655.35 in $.01 or $0-3,276.75 in $.05	80-95	Accumulation of Value Units 'This Week – 2'.
RtsUnitsWeek-	16	$0-655.35 in $.01 or $0-3,276.75 in $.05	96-111	Accumulation of Value Units 'This Week – 3'.

Table 23—AutoValue Product Object (AVPO) (continued)

Field	Size	Values	Pos.	Description
DACAVPO or CRCAVPO Data Authentication Code [DAC] or Cyclic Redundancy Check (CRC)	16	0-65535	112-127	DAC for authentication & error detection Or CRC for error detection only.
Total	**128**			

5.17.1 AVPO Extension

(User Information is not provided for either AVPO or AVPOX.)

5.18 AutoValue Product Object Extension

Table 24—AutoValue Product Object Extension (AVPOX)

Field	Size	Values	Pos.	Description
RtsUnitsMonthly	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	0-15	Accumulated of Value Units.
RtsUnitsBi-Month	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	16-31	Accumulated of Value Units.
RtsUnitsQuarterly	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	32-47	Accumulated of Value Units.
RtsUnitsYearly	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	48-63	Accumulated of Value Units.
RtsUnitsUD1	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	64-79	Accumulated of Value Units.
RtsUnitsUD2	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	80-95	Accumulated of Value Units.

Table 24—AutoValue Product Object Extension (AVPOX) (continued)

Field	Size	Values	Pos.	Description
RtsUnitsUD3	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	96-111	Accumulated of Value Units.
RtsUnitsUD4	16	$0 - 655.35 in 1c or $0 - 3,276.75 in 5c	112-127	Accumulated of Value Units.
Total	**128**			

NOTE—The "Units" values used above in the AVPO and AVPOX are representative of one example implementation and would be different for other currencies.

6. File Structure Design

While Part II of the Standard does not mandate a specific file structure for organizing the data objects and elements, there are a number of mandatory file structure design requirements. These are as follows:

— The DIO shall be placed in the core file (File 0).

— The PHPOX shall be placed in a separate file with a separate security key as it contains sensitive information about a patron and therefore should be protected by separate security keys from the other files.

— All PIOX must be declared in File-0 or the core file.

— The TPPO shall be placed in a separate file with a separate key set. This separate key set is to be disclosed only to the agencies and operators participating in the regional implementation.

— The SV&TPPOX must be placed within the same File as the core SV&TPPO.

— The ALRO and ALROX shall occupy a single dedicated file with a separate security write key to protect against tampering with its private information.

7. File Structure (Informative Only)

7.1 General

As described in Section 4.3, each object within the file structure (listed below) contains multiple data elements that define specific attributes to provide flexibility and comprehensive functionality within a PICC electronic transit application environment.

a) DIO

b) TAPO

c) PHPO

d) PIO

e) Product Object, (I.e. SV&TPPO, ALPO, P&TPO)

f) A&DVO

g) THO

h) Extension Objects

7.2 Applications and File Structure Implementation

The architecture is reasonably flexible in accommodating varying methods of file structure implementation. However, there are system level considerations that must be taken into account to prevent unnecessary memory consumption and, most importantly, unnecessary transaction time usage when implementing the file structure. The following issues should be considered:

a) Combined contents of core Elementary File (EF), File-0 as opposed to any other file

b) Level of patron, agency or regional fare product privacy and protection

c) Overall file-to-file access that propagates transaction time, where each of these files contains its own set of security keys (Minimizing read, write (update) and authentication cycles as much as possible)

One approach is to group the data and information necessary in preparation for a transaction as shown in Diagram 1 (File 0 Structure). Placed within the confines of (EF) File-0 are all of these objects, thus allowing a single file authentication for access and data retrieval. This enables the CID and hosting computer to retrieve and read the DIO and DIOX, TAPO, PHPO and PHPOX, A&DVO and THO, PIO and PIOX within one file authentication event. Diagram 1 below illustrates a sample File-0 implementation.

The Standard provides for additional capability that may be utilized within File-0, such as additional extensions to the PHPO besides the one listed in the example above. In this regard, the individual regional organization must decide what File-0 contains in total. It is recommended, however, that the contents of File-0 be given serious consideration as per the example File-0. As noted, 800 bytes are consumed by File-0 per the above example.

The full complement of A&DVOs and THOs supported by the Standard were selected to ensure that the depth of history trace-ability for data recovery be maximized to account for the high degree of system integrity required should thirteen or more fare products from multiple agencies be implemented. To ensure that the transit application and associated fare products are interoperable within a region it is necessary to agree on the depth of history trace-ability of File-0.

As with all systems, there are trade-offs with regard to data integrity, data recovery and overall PICC memory consumption. Even with several fare products supported on a PICC having only 2KB of data memory capability, a robust transportation application environment can be adequately supported.

7.3 Core Transit Application (EF) File-0

Elementary File-0 is the core Elementary File (EF) within the implementation (see Diagram 1). Since secure file access is time consuming, it is necessary to minimize the partitioning into separate files of Objects, especially those having separate key sets requiring additional file authentication. Having all of the necessary core objects placed into one core EF file ensures that a single file authentication event grants access to all the information needed during ISO 7816-4 “Select File” and “Read Binary” APDU commands. In the case where additional Product Objects are required beyond the five (5) allowed by the

core PIO, modification of the DIO "RtsPIOIndexSize" data would enable EF-0 to gain access to additional PIO Extension (s).

NOTE—The DIO, PHPO and TAPO should be written during card initialization and subsequent updates to these objects should be restricted to clerk-operated devices.

Directory Index Object (DIO)	16 Bytes
DIO Extension	16 Bytes
Transit Application Profile Object (TAPO)	16 Bytes
PICC Holder Profile Object (PHPO)	16 Bytes
Product Index Object (PIO) Current Core	16 Bytes
PIO Current Extension 1	16 Bytes
PIO Alternate Core	16 Bytes
PIO Alternate Extension 1	16 Bytes
PIO Current Extension 2	16 Bytes
PIO Current Extension 3	16 Bytes
PIO Alternate Extension 2	16 Bytes
PIO Alternate Extension 3	16 Bytes
Add & Deduct Value History Object (ADVO)	16 Bytes
ADVO 2-6	80 Bytes
Transaction History Object (THO) 1	16 Bytes
THO 2-16	240 Bytes
Product Object(s) – Regional and Other	16 Bytes
(R/PO) Core 1	16 Bytes
R/PO Extension 1	16 Bytes
R/PO Core & Extension 2-8	112 Bytes
Total	800 Bytes

Diagram 1—File 0 Structure

NOTE—In nearly all cases where a compliant Full-Featured PICC is used, there is a common file security function; If an EF with its own unique key is authenticated, access to as many reads as required within that EF are permitted (subject to the byte length restrictions of some ICs). If the write key is different from the read key, a separate authentication will normally be required for write functions. In the case of EF-0, a common combined read and write key (or a read and write key set) will be implemented. However, when information is required from a different EF and an authentication event is granted to gain access to that new EF, the original EF is no longer accessible. To regain access to the original EF, a successful authentication must be performed again. This is mentioned since each authentication event may require several milliseconds to perform.

7.4 PHPO Extension (EF) File

Even though packing all the necessary core objects into one EF file is an efficient method of implementation, there is no need to burden the EF, File-0 access time by including information of no immediate value. For example, if an implementation does not require an extension object, then there is no need to allocate data memory and read cycles to the attempted retrieval of such empty Object.

In addition, it should be recognized that some objects such as the PHPOX may contain personal patron information that must be securely protected. This is accomplished through the placement of this type of data within an Extension Object or Objects held in a separate EF that can be accessed only by use of an additional key set (See Diagram 2 PHPO Extensions).

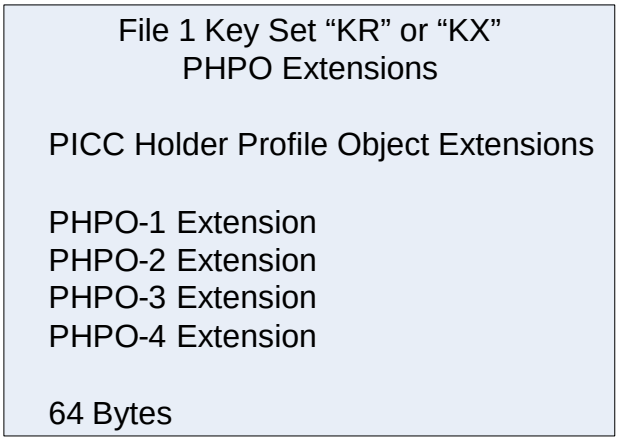


Diagram 2—File 1 Structure

7.5 T-Purse Object (EF) File

The T-Purse is actually a highly secure form of stored value that is designed for utilization within a regional environment comprised of multiple transit agencies requiring access to a single purse. Since data representing significant monetary value can be placed within the T-Purse, it is important that the T-Purse Object be as secure as possible, protected by a separate key (see Diagram 3). The structure of other files 3 to 6 is illustrated in Diagram 4.

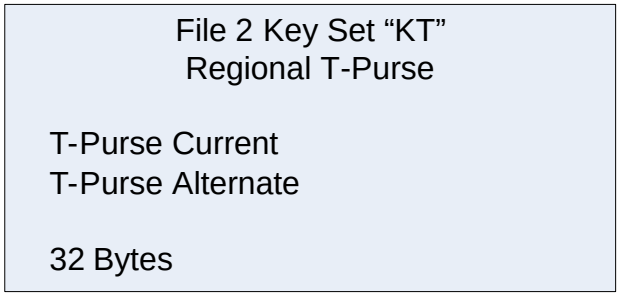


Diagram 3—File 2 Structure

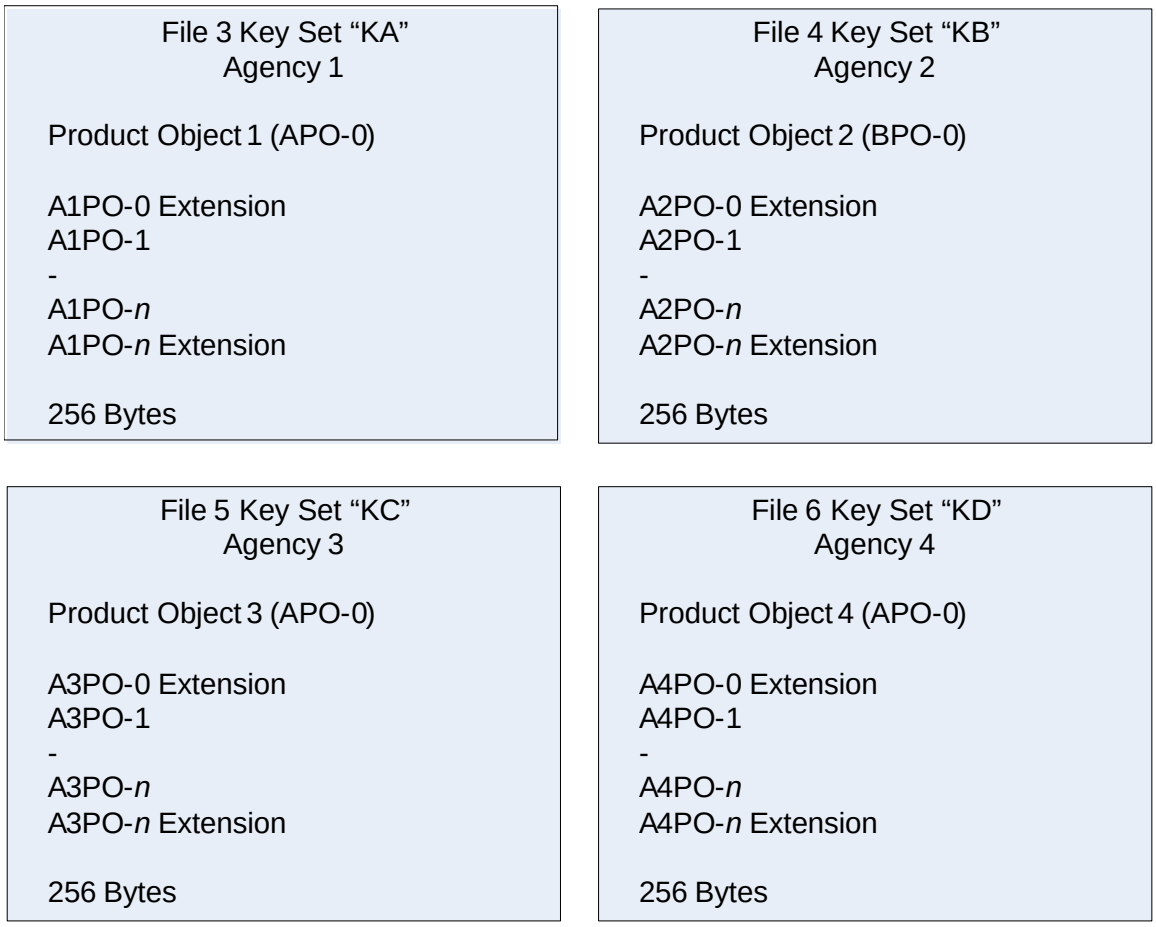


Diagram 4—Files 3-6 Structures

7.6 Account Linked Reference (EF) File

The ALRO is designed for utilization within a regional environment comprised of multiple transit agencies requiring access to an Account Linked product that is linked to a banking instrument such as a credit card. Since transaction data representing significant monetary value can be placed within the ALRO, it is important that the ALRO be as secure as possible, protected by a separate write key that should be owned by the entity responsible for maintaining this Object (See Diagram 5).

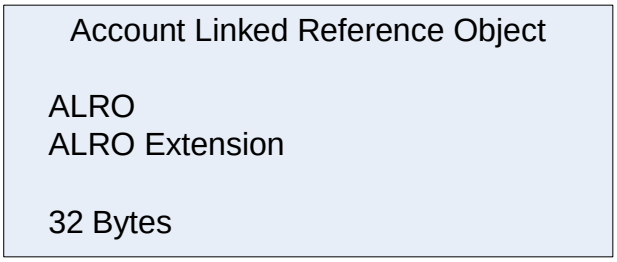


Diagram 5—Files 7 Structures