EMV®

QR Code Specification for Payment Systems (EMV QRCPS)

Consumer-Presented Mode

Version 1.0

July 2017

Legal Notice

The EMV® Specifications are provided “AS IS” without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party’s infringement of any intellectual property rights in connection with the EMV® Specifications.

QR Code is a registered trademark of DENSO WAVE.

Contents

Figures

Tables

1 Introduction

The EMV® QR Code Payment Specification (EMV QRCPS) Consumer-Presented Mode (this specification) defines the format, encoding, and decoding of the data "payload" of the consumer presented QR Code used to perform EMV based QR Code transactions.

1.1 Scope

The following are explicitly defined as in scope for this specification:

- The format of data encoded into the QR Code
- The QR Code Reader requirements to read and decode the data from the QR Code
- POI transaction processing on the data from the QR Code

The following are explicitly defined as outside the scope of this specification:

- Data sources used to populate the data in the QR Code
- Optical reader/scanner hardware requirements
- The mapping of data objects recovered from the QR Code into the online authorisation request message
- Payment systems rules for QR Code transactions
- Host processing of QR Code transactions
- Use of QR Code data for non-financial transactions

1.2 Normative References

Table 1.1 lists the documents referenced in this specification.

Table 1.1: References Materials

Short Form	Title and Description
[EMV]	EMV ICC Specifications for Payment Systems, Version 4.3, November 2011, and all published bulletins. Integrated Circuit Card Specifications for Payment Systems.
[EMV Bulletin No. 167]	EMV Specification Bulletin No. 167—Payment Account Reference (PAR)
[ISO 639-1]	Codes for the representation of names of languages—Part 1: Alpha - 2 Code. Note: This standard is updated continuously by ISO. Additions/changes to ISO 639-1:1988: Codes for the Representation of Names of Languages are available on: http://www.loc.gov/standards/iso639-2/php/code_changes.php
[ISO 7813]	ISO/IEC 7813. Identification Cards—Financial Transaction Cards

Short Form	Title and Description
[ISO 8583]	ISO/IEC 8583. Financial transaction card originated messages — Interchange message specifications
[ISO 7816-5]	ISO/IEC 7816-5. Identification Cards—Integrated Circuit Cards with Contacts—Part 5: Numbering System and Registration Procedure for Application Identifiers
[ISO 8825]	ISO/IEC 8825-1. Information technology—ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
[ISO 18004]	ISO/IEC 18004. Information technology—Automatic identification and data capture techniques—QR Code bar code symbology specification.
[RFC 3966]	RFC 3966. The tel URI for Telephone Numbers
[RFC 4648]	RFC 4648. The Base16, Base32, and Base64 Data Encodings
[RFC 6068]	RFC 6068. The 'mailto' URI Scheme

1.3 Notational Conventions

This chapter clarifies several terms and conventions used throughout this specification.

1.3.1 Abbreviations

Table 1.2: Abbreviations

Abbreviation	Description
ADF	Application Definition File
BER-TLV	Basic Encoding Rules- Tag Length Value
CVM	Cardholder Verification Method
CDCVM	Consumer Device Cardholder Verification Method
POI	Point of Interaction
PAR	Payment Account Reference
TRID	Token Requestor ID
UI	User Interface

1.3.2 Notations

Table 1.3: Value Representation and Notation

Value Representation	Notation
0 to 9	10 decimal digits. Decimal numbers are not enclosed in quotation marks.
'0' to '9' and 'A' to 'F'	16 hexadecimal characters, enclosed in straight quotation marks.
"Test@123"	Values that can be represented by the Common Character Set as defined in [EMV Book 4], enclosed in double straight quotation marks.

1.3.3 Requirement Terminology and Conventions

The following words are used often in this specification and have a specific meaning:

Shall

Defines a product or system capability which is mandatory.

May

Defines a product or system capability which is optional or a statement which is informative only and is out of scope for this specification.

Should

Defines a product or system capability which is recommended.

1.3.4 Requirement Numbering

Requirements in this specification are uniquely numbered with the number appearing next to each requirement.

For example:

6.1.1.1 The POI App shall parse the remainder of the chosen Application Template and store the recovered data objects for subsequent transaction processing. It should not parse the Application Specific Transparent Template (tag '63') and shall regard the contents of this template as a single data blob.

In addition to normative statements indicated by "shall", the requirement may include informative statements indicated by "should" or "may".

1.4 Data Objects

1.4.1 Data Object Presence

For the presence of data objects, the following notation is used:

- M: Mandatory—shall always be present.
- C: Conditional—shall be present under certain conditions.
- O: Optional—may be present.

For data objects residing in templates, the presence condition for a data object must be considered in conjunction with the condition of the template in which it resides.

1.4.2 Data Object Length

The length of all data objects is expressed in bytes, except where otherwise noted.

1.4.3 Data Object Format Conventions

Table 1.4: Data Object Format Conventions

Data Object	Format Conventions
an	Alphanumeric data objects contain a single character per byte. The permitted characters are alphabetic (a to z and A to Z, upper and lower case) and numeric (0 to 9).
ans	Alphanumeric Special data objects contain a single character per byte. The permitted characters and their coding are shown in the Common Character Set table in [EMV] Book 4 Annex B.
b	These data objects consist of either unsigned binary numbers or bit combinations that are defined elsewhere in this specification.
cn	Compressed numeric data objects consist of two numeric digits (having values in the range Hex '0'–'9') per byte. These data objects are left justified and padded with trailing hexadecimal 'F's.
n	Numeric data objects consist of two numeric digits (having values in the range Hex '0' – '9') per byte. These digits are right-justified and padded with leading hexadecimal zeroes. Other specifications sometimes refer to this data format as Binary Coded Decimal (BCD) or unsigned packed.

2 Overview to EMV® QR Code Payment

2.1 QR Code Payments

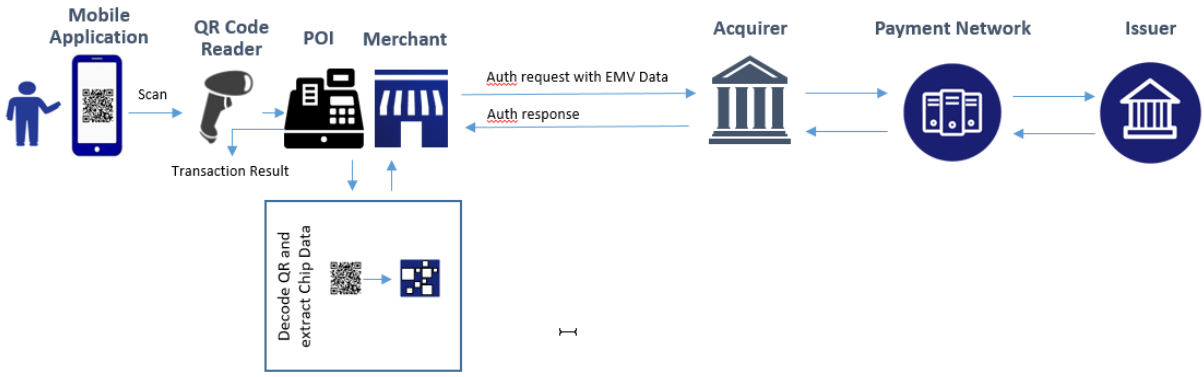
In an EMV Consumer-Presented Mode QR Code transaction, consumers can make purchases—using credentials associated with their EMV card and previously provisioned to their device—by selecting the QR option for payment within their mobile application, which will result in the display of the QR Code, and having that QR Code scanned at the time of payment to complete the transaction.

These transactions are always authorised online and given that the scanning of the QR Code is a one-way transfer of data from the consumer's device to the POI, the payload of the QR Code does not contain any data from the POI.

While out of scope of this specification, in the event that any cardholder verification is required, it is envisaged that it would be performed through CDCVM. Thus, unlike typical EMV transactions, the requirement for CVM is never communicated, or delegated to the POI. While specific markets may have POI CVM requirements in addition to any CDCVM that may have been performed, these POI CVM requirements are out of scope for EMVCo.

Figure 2.1 depicts the high-level solution architecture for QR Code payments processing. The components shown in the diagram are logical components that may map to different sets of implementation/physical components. The key objective of this diagram is to show the flow of QR Code data into the POI system and subsequently into the payment ecosystem. The diagram does not address other ecosystem components, such as provisioning of payment credentials and merchant value-added service data in the mobile application/wallet, user interface/interactions on the mobile device before and at the time of checkout; or mobile platform, etc.

Figure 2.1 QR Code Payment Architecture



The following describes the high-level functionality of various components of the QR Code processing architecture on the consumer device:

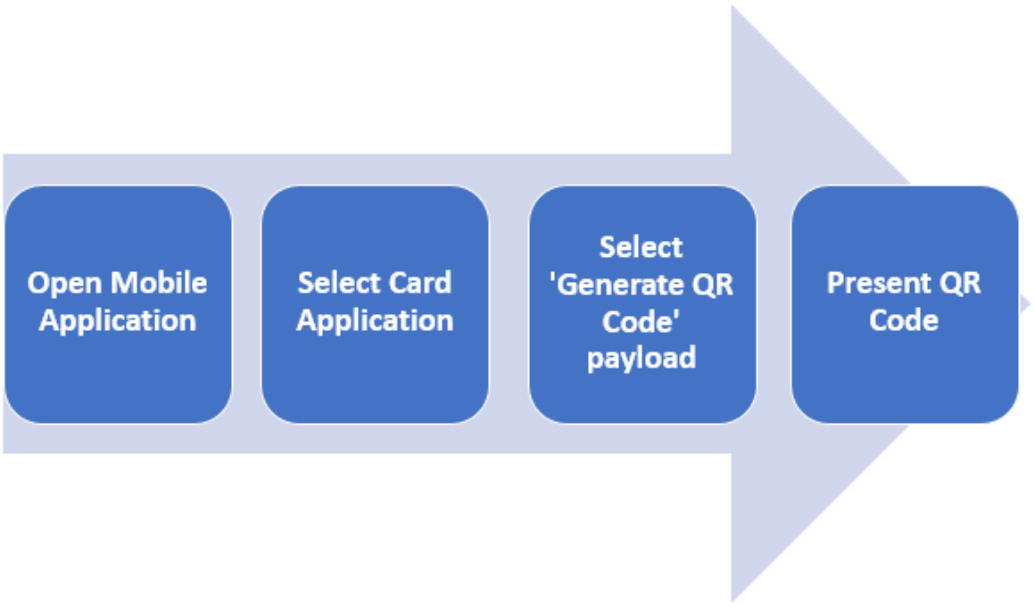
- **Mobile application/wallet**—A consumer-facing user interface (UI) application provided by the issuer, merchant, or a third-party and provisioned to the consumer's mobile device. It includes the functionality to encode the payment credentials based on this specification, and then displays the resulting QR Code.

If supported by the mobile application/wallet, the QR Code may also contain other consumer opt-in data, for example, for supporting additional services – these functionalities are not described in this specification.

The mobile application is responsible for providing a UI that will facilitate card application selection, perform any required CDCVM processing and rendering of the corresponding QR Code. Figure 2.2 illustrates the process for the consumer.

- **QR Code Payload**—The payload, consisting of payment token credentials and other data based on this specification, converted to base64 and encoded in a QR Code.

Figure 2.2 Consumer Device Functional View

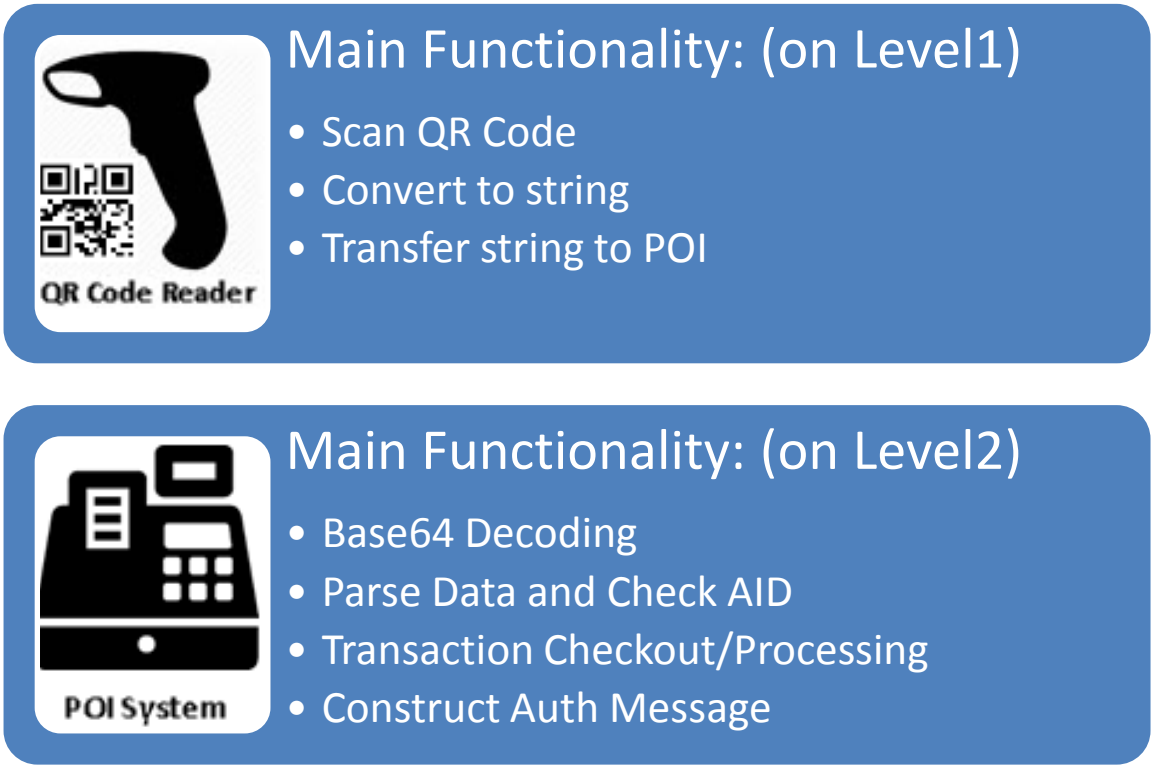


There are 2 logical components on the merchant device: the QR Code Reader and the POI Application.

- **QR Code Reader**—Scans the QR Code, decodes the QR Code and sends the data recovered to the POI System. This data constitutes the base64 encoded QR Code Payload.
- **POI Application**—The application developed by the POI vendor to process the base64 encoded QR Code payload defined in this specification. Its functions include base64 decoding, parsing the data, checking content and format and transaction processing.

Figure 2.3 illustrates their distinct functions.

Figure 2.3 Merchant Device Functional View



On the network the following are the roles as per the existing EMV infrastructure:

- Acquirer—The acquirer/processor processing the authorisation transaction received from the merchant.
- Payment Network—Responsible for transaction routing to the issuer and any corresponding tokenisation services and functions.
- Issuer—The issuer of the original payment card, that is used in the QR Code transaction.

3 EMV QR Code Payload Data Objects

This section defines the data objects that comprise the QR Code Payload.

3.1 QR Code Payload and Formatting

The data objects in the QR Code Payload are split into two main classes:

- Required data for the POI to process the payment and build the authorisation transaction to be forwarded to the Acquirer. For instance, the POI needs to be able to recognise data to present information to the consumer, print the receipt and recognise the ADF Name in order to correctly route the transaction. This class of data will be present in the Payload Format Indicator (tag '85'), Application template (tag ′61′) and/or the Common Data template (tag ′62′).
- Other data that the POI does not require for processing and is only forwarded to the acquirer. For instance, the Application Cryptogram ultimately used for transaction authorisation. This class of data may be present in the Application Specific Transparent Template (tag '63') or the Common Data Transparent Template (tag '64').

3.1.1 QR Code Payload

3.1.1.1 The data in the QR Code Payload shall consist of the BER-TLV coded data objects defined in Table 3.1.

Detailed descriptions of the data objects are provided in Annex A.

BER-TLV encoding is defined in [EMV] *Book 3 Annex B Rules for BER-TLV Data Objects*.

Table 3.1: QR Code Payload Data

Tag	Value			Length	Format	Presence
'85'	Payload Format Indicator			5	an	M
'61'	Application Template			var.	b	M
	'xxxx'	Additional BER-TLV coded data objects		var.	b	O
	'63'	Application Specific Transparent Template		var.	b	O
		'xxxx'	Additional BER-TLV coded data objects	var.	b	O
'61'	Application Template			var.	b	O
	'xxxx'	Additional BER-TLV coded data objects		var.	b	O
	'63'	Application Specific Transparent Template		var.	b	O
		'xxxx'	Additional BER-TLV coded data objects	var.	b	O
'62'	Common Data Template			var.	b	O
	'xxxx'	Additional BER-TLV coded data objects		var.	b	O
	'64'	Common Data Transparent Template		var.	b	O
		'xxxx'	Additional BER-TLV coded data objects	var.	b	O
'xx'	Other template			var	b	O
	'yy'	Another template or primitive data object		var	b	O

The Payload Format Indicator (tag '85') defines the QR Code format version and is the first data object of the payload. In this version of the specification, the Payload Format Indicator has the value "CPV01".

And the QR Code Payload contains either:

- 1 Application Template, or
- 2 Application Templates with 1 optional Common Data Template.

Other templates may also be present in the QR Code Payload.

Application Template(s) or the Common Data Template must precede the other templates, so that the Payload Format Indicator, Application Template(s) (if present) and the Common Data Template (if present) are the first data objects in the QR Code Payload.

As per BER-TLV encoding rules, templates and data within an Application Template or a Common Data Template can be in any order.

The "Additional BER-TLV coded data objects" in templates may include the data objects listed in Table 6.1. However, the same data objects must not appear in both template '61' and '62'.

The data objects included in the QR Code Payload should only include those critical for transaction processing and for the optimal consumer experience.

The following should be taken into consideration when determining the data objects to be included:

- Although there is no maximum cumulative size for the data objects in the QR Code Payload, QR Code Readers complying to this specification are only required to support recovery of 512 bytes of data from the QR Code.

  QR Code containing more than 512 bytes of data may not be supported by all QR Code Readers compliant to this specification. When assessing the cumulative size and ensuring that it will not exceed 512 bytes, the encoding described in section 3.2 must be taken into account.

- As the cumulative size of the data objects in the QR Code increases, the "QR Code-in-field" time also increases and the success rate of QR Code reading diminishes. The "QR Code-in-field" time is the duration in which the cardholder is required to keep the QR Code in the QR Code Reader's field for a successful read of the QR Code to occur.

3.2 QR Code Encoding

This section defines the steps for converting the QR Code Payload from binary data to a QR Code.

The QR Code is generated, encoded and displayed by the consumer device, however this exact procedure is out of scope of this specification. Statements in this section are informative descriptions and included to assist readers in understanding the functional requirements of QR Code Readers described in section 4.

The procedure to convert binary data into a QR Code can be accomplished by performing the following steps in the order shown.

1. Convert the binary data into a base64 encoded string as defined in [RFC 4648].
2. Convert the resultant base64 data string into a QR Code symbol as defined in [ISO 18004], using the following parameters and options:
   - **Mode:** use the default Extended Channel Interpretation (ECI) and Byte mode.
   - **Error Correction:** use Error Correction Level L, M, Q, or H and the value of p as defined in [ISO 18004] Table 9—Error correction characteristics for QR Code 2005.
   - **Symbol Version:** use the smallest QR Code version that accommodates the data.
   - **Data Masking:** perform data masking as defined in [ISO 18004].
   - **Orientation and Reflectance:** use normal orientation and normal reflectance arrangement.

4 QR Code Reader Requirements

This chapter defines the requirements of the QR Code Reader that shall be implemented in order to read an EMV QR Code. This specification uses the term "QR Code Reader" to indicate the optical scanner used to read and process the QR Code.

4.1 QR Code Reader

4.1.1.1 In order to read the EMV QR Code, the QR Code Reader shall be able to read QR Codes as defined in [ISO 18004] including the default ECI which is [ISO 8859-1].

Readers may support the following [ISO 18004] features:

- Micro QR Code
- Structured Append
- ECIs other than the default ECI
- Mirror Imaging/Orientation
- Reflectance Reversal

The QR Code Reader shall support recovery of at least 512 bytes from the QR Code. The QR Code Reader may support recovery of more than 512 bytes.

The QR Code read by the QR Code Reader is the base64 encoded QR Code Payload and shall be sent to the POI App.

Note: All other QR Code Reader and merchant point-of-sale (POS) hardware requirements are outside the scope of this specification.

Note: QR Code Reader behavior in response to any error that occurs during this processing is outside the scope of this specification.

5 POI App Requirements

This chapter defines the POI App processing necessary to decode and parse the base64 encoded QR Code Payload retrieved from the QR Code Reader.

5.1 EMV QR Code Processing Requirements

This section defines the procedure performed by POI App to process the EMV QR Code and retrieve the data encoded therein.

5.1.1 Decode the QR Code Data

5.1.1.1 If the data received from the QR Code Reader starts with "hQVDUFY", then the POI shall perform base64 decoding as defined in [RFC 4648] and shall subsequently parse the data.

The POI shall support at least 512 bytes of QR Code Data and may support more.

Note: If the data received does not start with "hQVDUFY", then the QR Code is not encoded per this specification and POI processing of the QR Code is outside of the scope of this specification.

5.1.2 Verify EMV QR Code

5.1.1.2 If any of the following [in the order presented] is true, then the data is not encoded per this specification and subsequent processing is outside the scope of this specification:

- The data received from the QR Code Reader is not base64 encoded.
- Following base64 decoding, the resulting binary data is not BER-TLV coded.
- The first BER-TLV encoded data is other than the Payload Format Indicator defined in Section 3.1.

5.1.3 Parsing and Format Validation

5.1.1.3 The POI shall parse the BER-TLV data to retrieve the EMV QR Code Payload defined in Section 3.1.

If the value of the Payload Format Indicator (tag '85') is not "CPV01", or the data does not contain a template with a tag of '61' (Application Template), then the POI App shall indicate that an error has occurred. Subsequent processing is outside the scope of this specification

The QR Code Data may additionally include other templates not defined in this specification. The POI may check that these templates are correctly BER-TLV coded, but all other processing of these templates is outside the scope of this specification.

5.1.4 Application Template(s)

5.1.1.4 For each Application Template (tag '61'), the POI App shall examine the ADF Name (tag '4F') to determine whether the corresponding application (identified by an AID) is supported by the POI App.

If the template does not contain a data object of '4F' (Application Definition File (ADF) Name), then the POI App shall stop processing the current template and continue with the next template (if any).

If the length of the ADF Name is less than 5 or greater than 16, then the POI App shall stop processing the current template and continue with the next template (if any).

If the ADF Name matches an AID supported by the POI App, then the respective Application Template is eligible for this transaction.

If the ADF Name does not match an AID supported by the POI App, the POI App shall ignore the Application Template and its contents.

The ADF Name (tag '4F') matches an AID in the POI App if:

- the ADF Name has the same length and value as the AID.
- the ADF Name has a length greater than the length of the AID and the value of the AID equals the value of the left most bytes of the ADF Name.

5.1.5 No Eligible Application Template

5.1.1.5 If no Application Template within the EMV QR Code is eligible for this transaction, then subsequent processing is outside the scope of this specification.

5.1.6 Multiple Eligible Application Templates

5.1.1.6 If two (2) Application Templates within the EMV QR Code are deemed to be eligible, then the POI App shall support a configurable option allowing the merchant to determine which of the eligible Application Templates to use for the transaction.

Once this determination has been made, the POI App shall ignore the content of the Application Template that was not chosen.

5.1.7 Chosen Application Template

5.1.1.7 The POI App shall parse the remainder of the chosen Application Template and store the recovered data objects for subsequent transaction processing. It should not parse the Application Specific Transparent Template (tag '63') and should regard the contents of this template as a single data blob.

5.1.8 Common Data Template

5.1.1.8 If the Common Data Template (tag '62') is present, then the POI App shall parse the Common Data Template and store the recovered data objects for later transaction processing. It should not extract data objects from the Common Data Transparent Template (tag '64') and shall regard the contents of this template as a single data blob.

5.1.9 Combining Templates

5.1.1.9 The POI App shall create a set of data from the chosen Application Template and, if present from the Common Data Template, namely:

- **POI Data**. This will include every data object excluding the content of the Application Specific Transparent Template (tag '63') and, if applicable, the Common Data Transparent Template (tag '64').
- **Transparent Data**. This will include the content of the Application Specific Transparent Template (tag '63') and, if applicable, the Common Data Transparent Template (tag '64').

5.1.10 Duplicate POI Data

5.1.1.10 When creating the POI Data as described above, the POI App shall check for duplicate data objects. This only applies to POI Data and does not apply to any content within the Transparent Data.

If there is more than one occurrence of a single primitive data object in the POI Data, then the POI App shall indicate that an error has occurred. Subsequent processing is outside the scope of this specification.

Note: While the same primitive data objects may have been present in the case of two Application Templates, as per 5.1.6, the discarded Application Template and all data therein is ignored.

5.1.11 Unrecognised Data Objects

5.1.1.11 The POI App shall ignore but not discard unrecognised data objects (data not defined in this specification) from the POI Data.

The POI App shall be capable of accepting additional templates not explicitly shown in the QR Code Data in Table 3.1. These templates and the data within may not be related to payment and the processing of such is outside the scope of this specification.

Note: The majority of the data objects defined in this specification are used in support of financial transactions (for example, to populate the authorisation message), but additional data objects may be present in the QR Code Payload in support of additional proprietary services that are outside the scope of this specification.

5.1.12 Data Processing

5.1.1.12 After the consistency of the data has been checked (5.1.1 to 5.1.10), the POI App continues with transaction processing.

6 Transaction Processing

After the POI has processed the QR Code data payload, it has two sets of data: the POI Data and the Transparent Data.

The POI uses the POI data for transaction processing and both sets of data to build the online messages. The POI does not use the Transparent Data for transaction processing.

Transaction processing includes the following:

- Check whether the payment product is supported
- Process Track 2 Equivalent Data (tag '57')
- Display of POI messages
- Print receipt
- Determine format of Transparent Data
- Perform merchant specific processing

Transaction processing is based on the data in Table 6.1 that also indicates requirements for presence of the data.

Table 6.1: List of POI Data

Tag	Data Object	Presence	Condition
'4F'	Application Definition File (ADF) Name	M	
'50'	Application Label	O	
'57'	Track 2 Equivalent Data	C	Either Track 2 Equivalent Data must be present or Application PAN must be present.
'5A'	Application PAN	C	
'5F20'	Cardholder Name	O	
'5F2D'	Language Preference	O	
'5F50'	Issuer URL	O	
'9F08'	Application Version Number	O	
'9F19'	Token Requestor ID	O	
'9F24'	Payment Account Reference	O	
'9F25'	Last 4 Digits of PAN	O	

Except for the Issuer URL (tag '5F50'), for the definition of these data objects, please refer to [EMV] Book 3. For the definition of the Issuer URL (tag '5F50'), please refer to Table 6.3.

6.1 Check Whether Payment Product is Supported

6.1.1.1 Based on the Application Definition File (ADF) Name (tag '4F'); the POI matches the ADF against the list of supported AIDs, as described in 5.1.4 Application Template(s).

6.2 Track 2 Equivalent Data: Data Mapping

6.1.1.2 If the PAN (tag '5A') is absent, then the POI must extract the PAN from Track 2 Equivalent Data.

Track 2 Equivalent Data (tag '57') contains the data objects of the track 2, in accordance with [ISO 7813], excluding start sentinel, end sentinel, and LRC. The Track 2 Equivalent Data has a maximum length of 19 bytes (38 nibbles) and is made up of the following sub-fields:

Table 6.2: Track 2 Equivalent Data (Tag '57')

Data Field	Length	Format
Primary Account Number	var. up to 19 nibbles	n
Field Separator of 'D'	1 nibble	b
Expiration Date (YYMM)	4 nibbles	n
Service Code	3 nibbles	n
Discretionary Data	var.	n
Padded with 'F' if needed to ensure whole bytes	1 nibble	b

6.3 POI Messages

6.1.1.3 If the Language Preference (tag '5F2D') is present in the QR Code Data, and if one of the listed languages is supported by the POI, then the supported language with the highest preference shall be used when displaying messages to the cardholder.

6.4 Print Receipt

6.1.1.4 If the POI prints the last four digits of the PAN on the receipt and the Last 4 Digits of PAN (tag '9F25') is present in the QR Code Data, then the POI shall use the value of the Last 4 Digits of PAN (tag '9F25') when printing the last four digits of the PAN on the receipt.

If the POI supports receipt printing, the following data objects may be printed on the receipt:

- Application Label (tag '50')
- Cardholder Name (tag '5F20')

If the POI supports electronic means for providing receipts, such as email or SMS, then the POI may analyse the Issuer URL (tag '5F50') for supporting data. For this version of the specification, the definition and the format of the Issuer URL (tag '5F50') is as identified below in Table 6.3.

Table 6.3: Issuer URL (Tag '5F50')

Tag	Data Object	Format	Length	Description
'5F50'	Issuer URL	ans	Var.	The value of the Issuer URL contains customer information for electronic receipt delivery with the syntax defined by a standard URI scheme. Supported URI schemes are: • Phone number, defined in [RFC 3966] • Email address, defined in [RFC 6068]

The following are examples of the Issuer URL:

- If the value is a phone number: tel:+1-234-567-8910
- If the value is an email address: mailto:example@emvco.com

6.5 Determine Format of Transparent Data

6.1.1.5 Based on the combination of the AID matching the ADF Name (tag '4F') and the Application Version Number (tag '9F08'), the POI determines the convention for mapping data listed in the Transparent Data to an online authorisation message. If the Application Version Number (tag '9F08') is absent, the POI shall use '00 10' (version 1.0) as the default value for the Application Version Number.

6.6 Perform Merchant Specific Processing

6.1.1.6 The PAR (tag '9F24') allows acquirers and merchants to link transactions, whether tokenised or not, that are associated to the same underlying account. The POI shall be able to recognise the PAR, but use of the PAR is out of scope of this specification.

Processing of the Token Requestor ID (TRID, tag '9F19') is out of scope of this specification, however the POI shall be able to recognise the data object.

7 Authorisation

EMV QR Code payment transactions are online authorised.

This specification assumes that:

- Merchants and acquirers support full chip data in order to support EMV QR Code transactions.
- Issuers (or any processing entity on their behalf) are able to process EMV QR Code transactions and validate the application cryptogram included in the chip data.

For general guidance on the network message and the data objects, refer to the 'Acquirer Interface' section and 'Annex C - Example Data Element Conversion' in [EMV] Book 4.

Data objects included in the authorisation message for a QR Code transaction, and the origin and the values of these data objects, may differ from those for a chip transaction. For example:

- The Unpredictable Number (UN, tag '9F37') may come from the QR Code and therefore be card sourced while for a chip transaction it would be terminal sourced.

Mapping of the POI Data and the Transparent Data to an online authorisation request and the default values requirements are payment system specific and therefore outside of the scope of this specification. Please contact the respective payment systems for guidance.

Annex A Data Objects Dictionary

This appendix defines the data objects that may be included in the QR Code data.

A.1 Data Object Descriptions

Table A.1 lists the data objects used in this specification.

Name, Format, Tag, and Length

The *Name* column lists the name of the data object and also includes the following:

- Format (F) of the data object. The supported formats are as follows:
  - an (alphanumeric)
  - ans (alphanumeric special)
  - b (binary or bit string)
  - cn (compressed numeric)
  - n (numeric)
- Tag (T) of the data object in hexadecimal.
- Length (L) of the data object in bytes.

Table A.1: Data Objects

Name (Format; Tag; Length)	Description	Values
Application Dedicated File (ADF) Name F: b T: '4F' L: 5–16	Identifies the application as described in [ISO 7816-5]. The ADF Name may also be referred to as the Application Identifier (AID). The POS system shall maintain a list of applications supported by the POS system identified by their AIDs.	Refer to Table 6.1/[EMV] Book 3
Application Label F: ans* T: '50' L: 1-16	Mnemonic associated with AID according to [ISO 7816-5].	Refer to Table 6.1 /[EMV] Book 3 * Special characters limited to space.
Application PAN F: cn T: '5A' L: var. up to 10	Valid cardholder account number.	Refer to Table 6.1/[EMV] Book 3

Name (Format; Tag; Length)	Description	Values
Application Specific Transparent Template F: b T: '63' L: var.	Contains the application specific data that is transparent to the POI App, where the application is identified by the ADF name.	Refer to Table 3.1
Application Template F: b T: '61' L: var.	Contains the application specific data where the application is identified by the ADF name.	Refer to Table 3.1.
Application Version Number F: b T: '9F08' L: 2	Version number assigned by the payment system for the application.	If this data is absent, the POI uses '00 10' (version 1.0) as the default value
Common Data Template F: b T: '62' L: var.	Contains the common data that is applicable to the POI App, for the application(s) in Application Template(s).	Refer to Table 3.1
Common Data Transparent Template F: b T: '64' L: var.	Contains the common data that is transparent to the POI App, for the application(s) in Application Template(s).	Refer to Table 3.1
Cardholder Name F: ans T: '5F20' L: 2–26	Indicates cardholder name according to [ISO 7813].	Refer to Table 6.1/[EMV] Book 3

Name (Format; Tag; Length)	Description	Values
Issuer URL F: ans T: '5F50' L: var.	Contains customer information for electronic receipt delivery with the syntax defined by a standard URI scheme.	Refer to Table 6.1/identified below in Table 6.3. Table 6.3
Last 4 Digits of PAN F: n T: '9F25' L: 2	Represents the last four digits of the underlying PAN affiliated with the Payment Token.	Refer to Table 6.1.
Language Preference F: an T: '5F2D' L: 2–8	1 - 4 languages stored in order of preference, each represented by 2 alphabetical characters according to [ISO 639-1]	Refer to Table 6.1/[EMV] Book 3.
Track 2 Equivalent Data F: b T: '57' L: var. up to 19	Contains the data objects of the Track 2 according to [ISO 7813], excluding start sentinel, end sentinel, and LRC, as follows: • Primary Account Number: numeric, var. up to 19 digits • Field Separator ('D'): binary • Expiration Date (YYMM): numeric, 4 digits • Service Code: numeric, 3 digits • Discretionary Data: numeric, var. • Pad with 'F' if needed to ensure whole bytes: binary	Refer to Table 6.1/Table 6.2
Token Requestor ID F: n T: '9F19' L: 6	Uniquely identifies the pairing of Token Requestor with the Token Domain.	Refer to Table 6.1
Payload Format Indicator F: an T: '85' L: 5	Defines the version of the QR Code. The first 3 characters are always “CPV” (Consumer Presented Version) and the last two characters must be decimal digits indicating the version of the payload format.	"CPV01"

Name (Format; Tag; Length)	Description	Values
Payment Account Reference F: an* T: '9F24' L: 29	Uniquely identifies the PAN to which a payment token is associated, as defined in [EMV Bulletin No. 167].	Refer to Table 6.1 * Permitted characters are alphabetic *upper* case and numeric.

Annex B Examples

This appendix provides examples for converting binary data to and from a QR Code as defined in this specification.

Note: The data object values provided in the examples below are for illustrative purposes only and may not be indicative of typical values for each data object.

B.1 Example 1

Binary Data (shown as hex bytes):

```
85 05 43 50 56 30 31
61 1A
        4F 07 A0 00 00 00 55 55 55
        57 0F 12 34 56 78 90 12 34 58 D1 91 22 01 12 34 5F
```

Base64 Data:

```
hQVDUFYwMWEaTwegAAAAVVVVVw8SNFZ4kBI0WNGRIgESNF8=
```

QR Code:

B.2 Example 2

Binary Data (shown as hex bytes):

```
85 05 43 50 56 30 31
61 13
        4F 07 A0 00 00 00 55 55 55
        50 08 50 72 6F 64 75 63 74 31
61 13
```

```
        4F 07 A0 00 00 00 66 66 66
        50 08 50 72 6F 64 75 63 74 32
62 49
        5A 08 12 34 56 78 90 12 34 58
        5F 20 0E 43 41 52 44 48 4F 4C 44 45 52 2F 45 4D 56
        5F 2D 08 72 75 65 73 64 65 65 6E
        64 21
                9F 10 07 06 01 0A 03 00 00 00
                9F 26 08 58 4F D3 85 FA 23 4B CC
                9F 36 02 00 01
                9F 37 04 6D 58 EF 13
```

Base64 Data:

```
hQVDUFYwMWETTwegAAAAVVVVUAhQcm9kdWN0MWETTwegAAAAZmZmUAhQcm9k
dWN0MmJJWggSNFZ4kBI0WF8gDkNBUkRIT0xERVIvRU1WXy0IcnVlc2RlZW5kIZ8QBwY
BCgMAAACfJghYT9OF+iNLzJ82AgABnzcEbVjvEw==
```

QR Code:

***** END OF DOCUMENT *****