https://verifone.cloud/docs/sca-functional-specification/protocol_spec/message_format
Updated: 20-May-2025

MESSAGE FORMAT

All SCA messages (requests and responses) should be fully formed XML documents compliant to the 1.0 specification. See http://www.w3.org/TR/REC-xml/ for the full specification.

Example

```
<DOCUMENT_ELEMENT>
    <ELEMENT />
    <ELEMENT>TEXT CONTENT</ELEMENT>
    <ELEMENT>TEXT CONTENT</ELEMENT>
    <ELEMENT>
    <ELEMENT>TEXT CONTENT</ELEMENT>
    </ELEMENT>
</DOCUMENT_ELEMENT>
```

Empty elements are represented by form <ELEMENT />, not <ELEMENT></ELEMENT>.

Requests to SCA

All request messages should have a document element of TRANSACTION. Each field name should be represented by a child element's tag name and the text content should represent that field's value. The following elements are required for all requests, except for SECURITY functions.

Note

XML Request messages must be entirely upper case – with the exception of customizable text values. Do NOT declare the XML version in the message. Best practice is to send the entire message in one packet of data.

Field	Value	Comments
FUNCTION_TYPE	Ex: SESSION	Type of function performed. Function types group COMMANDs into logical categories.
COMMAND	Ex: START	Command requested. COMMANDs that are not directly handled by Point will be passed to the Point platform for processing.

Field	Value	Comments
COUNTER	Ex: 100	Used to authenticate the POS. Each COUNTER should be higher than the last one. Min value:1 and Max value: 4294967295
MAC		Used to authenticate the POS. MAC value of the COUNTER using the decrypted MAC_KEY returned from the REGISTER command. This value is Base64 encoded.
MAC_LABEL	Ex: REG1	Used to authenticate the POS. Associated label that tells the device which MAC_KEY to use to decrypt the value of MAC.

Example:

```
<TRANSACTION>                                          // Start of TRANSACTION
    <FUNCTION_TYPE>SESSION</FUNCTION_TYPE>             // FUNCTION_TYPE = SESSION
    <COMMAND>START</COMMAND>                           // COMMAND = START
    <COUNTER>1</COUNTER>                               // COUNTER = 1
    <MAC> … </MAC>                                     // MAC = …
    <MAC_LABEL>REG2</MAC_LABEL>                        // MAC_LABEL = REG2
</TRANSACTION>                                         // End of TRANSACTION
```

Responses from Point

All responses will have a document element of RESPONSE. With the exception of reports, each RESPONSE element has at least the following fields.

Field	Value	Comments
TERMINATION_STATUS	Ex: SUCCESS	End status of the operation.
RESULT_CODE	Ex: -1	End result of the operation.
RESULT	Ex: OK	Human readable result.
RESPONSE_TEXT	Ex: OK	Specific details of the result. This may contain actual verbiage from the processing platform and can vary from platform to platform.

Each field name should be represented by a child element’s tag name and the text content should represent that field’s value.

Example:

```
<RESPONSE>
// Start of RESPONSE
    <TERMINATION_STATUS>SUCCESS</TERMINATION_STATUS>
// TERMINATION_STATUS=SUCCESS
    <RESULT_CODE>-1</RESULT_CODE>
// RESULT_CODE=-1
    <RESULT>OK</RESULT>
```

```
// RESULT=OK
    <RESPONSE_TEXT>SESSION started</RESPONSE_TEXT>
// RESPONSE_TEXT=SESSION started
<RESPONSE>
                                                        // End of RESPONSE
```

All PAYMENT non-follow-on transaction responses (e.g., AUTHORIZATION or CAPTURE) will include both a
TROUTD and CTROUTD in the RESPONSE element unless transaction was processed offline or ERROR.

Note

SCA supports the use of CTROUTD in follow on transactions.

Field	Value	Comments
CTROUTD	Ex: 123	The CTROUTD parameter is a sequence number for PAYMENT transactions (always enabled) that is generated per Client ID. Each Client ID has its own CTROUTD sequence counter. NOTE: Differs for Enterprise Direct implementations: see Appendix C
TROUTD	Ex: 14967	The TROUTD parameter is a sequence number for PAYMENT transactions (always enabled) that is generated per transaction. NOTE: Differs for Enterprise Direct implementations: see Appendix C

Note

Classic Implementation: The TROUTD has a life of 10 months from when it was originally processed.

Note

Classic Implementation: CTROUTD values will grow at a slower rate than the TROUTD value. Thus, the CTROUTD value may consist of fewer digits than the TROUTD, making it more convenient for manual entry.

TROUTD and CTROUTD values are unique. The CTROUT is unique to a given client. The CTROUTD value will be used in place of its associated TROUTD value in follow on transactions.

Note

When using the Point Gateway, CTROUTD has a life of 10 months from when it was originally processed. The CTROUTD is unique to a given client.

PAYMENT transaction responses may conditionally contain any of the following in the RESPONSE element.

Field	Value	Comments
LPTOKEN	Ex: 12	LPTOKEN will be included in the Point response when present in the gateway response. When enabled, Verifone provides an LP Token (LPTOKEN) in the response of each unique card number processed throughout the UGP Gateway. The LP Token is a non-sensitive unique number assigned to each unique card number processed with the UGP gateway. This value will automatically increment by one for each unique card number. NOTE: Applies to Classic implementation
PAYMENT_TYPE	Ex: CREDIT	PAYMENT_TYPE will be included in the Point response when present in the UGP response.
CARD_TOKEN	Ex: 7987654321098765	NOTE: For use with host based processors supporting card based token implementations. Refer to Two Way Card Token section for more details.
BANK_USERDATA	Ex: 7987654321098765	Conditional

Follow-on Transactions

To allow an integrated application to process follow on transactions effectively, it is recommended that the application:

1. Store the TROUTD and CTROUTD values
2. Store the status of the transaction:
    1. Whether it has been voided
    2. Whether it has been completed
    3. Whether the gratuity has been added
    4. Whether it has been settled (and is not eligible for further follow on transactions)

If the application will store this information, lists or menu options should be provided by the application that allow merchants to perform follow on transactions easily.
Alternatively, with Classic implementations, the merchant can use the Store Portal (virtual terminal) to process follow on transactions. The Store Portal allows all follow on transactions to be processed, such as Voids, Credits, Completions, and Tips.

The successful response varies depending on card type, but the failure response will always be either ERROR or DECLINED.

- CAPTURED: This response indicates the transaction is eligible for settlement.
- APPROVED: This response indicates the cardholder's available credit has reduced by the amount of the AUTHORIZE transaction, but the transaction will not be eligible for settlement until it is followed by a CAPTURE transaction.
- ERROR: This response typically indicates a communication problem, but it can appear at other times.
- DECLINED: This response indicates that transaction was sent to the payment processing company, but they declined it.
- DECLINED DUPLICATE TRANSACTION: This response indicates the transaction was declined by the Point platform as a duplicate transaction. The merchant has the option to resubmit the transaction and force the Point platform to send the duplicate transaction to the payment processor.

Refer to Result/Error Codes for more information about the required RESPONSE elements.

Two Way Card Token

Note

This section is only applicable to host based processors/gateways supporting card based token implementations. Includes Classic implementations.

PAYMENT transaction responses may conditionally contain a two way token in the RESPONSE element (in a non-token transaction). PAYMENT transaction requests may conditionally contain a token in the REQUEST element (when token is known).

Field	Value	Comments
CARD_TOKEN	Ex: 7987654321098765	CARD_TOKEN Ex: 7987654321098765 The Card Token parameter is returned in AUTH, CAPTURE, and CREDIT (Refund) SCI response messages to the POS. It can also be returned in most GIFT administrative transactions. The processor/gateway will give the token to the payment device and SCA will return CARD_TOKEN in the SCI response message to the POS. The card token is processor-based or gateway-based and can represent a unique card.

Field	Value	Comments
CARD_TOKEN	Ex: 7987654321098765	CARD_TOKEN can be used in subsequent AUTH, CAPTURE, or CREDIT requests to represent the card.