SVS

Web Services Programmer's Guide

Version 3.7

May 2017

Document Information

Revision History

Date	Version	Description	Author
05/20/05	1.0	Revision and re-title of Gift Card XML Spec	John McConville, SVS Chris Carrico, SVS Tim Roberts (Editor), SVS
09/15/06	1.1	Revision – added Return Codes	Tim Roberts, SVS
10/15/07 1/15/08 6/30/08	1.2 1.21 1.22	Revision – changed URL addresses. Logo change. No changes to content. Updates to tables in Appendix B.	Katy Nugent, SVS
9/30/08	2.0	Major revision of Web Services Programmers Guide.	Brad Stafford, SVS Devin Dowell, SVS Tania Cambron, SVS Katy Nugent
4/17/09	2.1	Added Enhanced Balance Inquiry.	Cognizant
11/5/09	2.2	Updated Response Types paragraph.	Katy Nugent (for International)
4/5/10	2.3	Logo change. No changes to content.	Laura Jones
7/15/10	2.4	Added response codes 35, 36, 37, 38, and 42. Added information to Enhanced Balance Inquiry section.	Laura Jones
8/13/10	2.5	Updated Time element in EnhancedBalanceInquiryResponse table.	Katy Nugent
11/25/11	2.6	Updated SOAP and WSDL information: • soap version • security version • CERT Link • Prod Link • Date Type Input String • Sample Request • Sample response Added new messages for Loyalty: • Loyalty EvaluatePurchase	Rakesh Kumar, Cognizant Joanne Weick, SVS

		• Loyalty PreAuthPurchase • Loyalty CompletePurchase Updated Transaction Glossary Updated Message Element Glossary	
1/26/12	2.7	Updated WSDL information to include new merchant fields for EBI response and added EoV fields.	Thomas Ehrler
8/3/2012	2.7	Fixed some spelling and formatting errors – no new content added	Thomas Ehrler
2/7/2013	2.8	Added new secondary security code details, updated version, year	Thomas Ehrler
3/1/2013	2.9	Added additional note to transaction sets, updated Verisign certificate, updated response codes	Tania Cambron
4/29/2013	3.0	Revised SOAP information that was added incorrectly in revision 2.6	Thomas Ehrler
11/12/2014	3.1	Added text detailing SVSXML DNS and corresponding IP ranges.	Thomas Ehrler
3/18/2014	3.2	New WSDL version that includes all the existing services along with Enable account transaction, Sku Processing,Promotional Code Processing	Ambika Vivekanandan
8/01/2015	3.4	New WSDL which includes Issue Virtual Third Party	Mahesh Kundal
2/15/2016	3.5	Update Boolean values to lowercase true/false	Michael Hasty
8/23/2016	3.6	Remove Loyalty	Michael Hasty
3/10/2017	3.7	Added Coupon Code and Card Type support	Mahesh Kundal
5/15/2017	3.7	Added Campaign Code	Michael Hasty

Disclaimer

The information furnished herein by Stored Value Solutions is proprietary and confidential and is distributed to clients and authorized business partners for their exclusive use in operating its Cash Card programs, and shall not be duplicated, published, or disclosed in whole or in part without the prior written permission of SVS.

Contents

Overview

Welcome to Web Services

Stored Value Solutions provides multiple Application Programming Interfaces (API) by which a client can process Gift Card Transactions. ISO8583 Standards, Dial Standards and Web Services are just some of the supported APIs. This document has been designed to help guide a client through the design and implementation of the SVS Web Services API**.**

Intended Audience

This document is written for systems analysts and programmers already familiar with Web Services application programming standards such as the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL), and XML Schema Definition (XSD) language.

Organization of Document

Basics of Stored Value Solutions Transaction Processing presents an explanation of the logic behind SVS Transaction Processing. The focus will be on the general concepts that are common to all of the APIs provided by SVS.

Stored Value Solutions Web Services describes the architecture of the SVS Web Services, its version and security features, the SOAP and WSDL implementation, and the two distinctive processing styles, Dup-Check vs. Reversal, that are supported.

Appendix A: Soap Envelope Example provides two examples of SOAP envelopes. (Request and SVS response)

Appendix B: SVS Web Services API Reference provides a detailed breakdown of the elements that you will use to build the SOAP messages. Data types, size limitations, and example values are provided in this section.

Appendix C: Diagrams provides diagrams of the Failed Transactions Dup-check, Failed Transaction Logic Reversal logic, and Pre-Authorization and Pre-Authorization Complete transactions.

Appendix D: Transaction Glossary provides an overview of commonly used terminology.

Appendix E: Message Element Glossary provides an overview of messaging terminology.

Basics of SVS Transaction Processing

In this section we will discuss the basics of SVS Transaction Processing. The purpose of this section is to familiarize you with SVS terminology, SVS Card components, and request/response interaction with the SVS Card. The topics covered in this section are:

- The SVS Card
- SVS Card Types
- Interacting with your SVS Card (Request/Response)
- Request Types
- Response Types

The Stored Value Solutions Card

The one common element that all SVS clients share is the SVS Card. A *SVS Card is an individual database record stored on the SVS Transactional Database*. The SVS Card record contains many fields. Below are some of the fields we will use in further discussion.

- Card Number – A unique identifier of the SVS Card
- Card Pin Number – A value used to restrict access to the SVS Card. This field is divided into two subfields; the first 4 digits are the secondary security code (SSC) to be used with bar coded accounts. The second subfield is a 4 digit PIN number used for any card not present transaction or when the magnetic strip is not available to add extra security for the card account. **Each Sub-field should be right justified left zero filled**.
- Card Status – An identifier of the state of the SVS Card
- Card Working Balance – Amount of monetary value associated to the SVS Card
- Card Out Auth Amount – Amount of monetary value being held on the SVS Card
- Card Available Balance – Amount of monetary value available for immediate usage on the SVS Card. This is derived from (Card Working Balance – Card out Auth Amount).EOV / End of Validity Date Day as of which the Card Deactivate. SVS returns EOV based on the request regardless of the feature turned on or not. Enforcing the EOV is dependent on if you contact SVS to have that turned on.

Stored Value Solutions Card Types

As a client of SVS you have been assigned one or more BIN ranges. *A SVS BIN Range is defined as a range of card numbers reserved on the SVS Transactional Database.* Even though a client has been assigned a BIN range, the card records do not exist on the SVS Transactional Database until the card data is created. The SVS Card Type is determined at the time the card data is created. The following are the three most common SVS Card Types:

- Customer-Valued Cards – A customer-valued card has a status of active with a working balance of $0.00 when initially created.
- Pre-Valued Cards – A pre-valued card has a status of inactive with a working balance of greater than $0.00.
- Virtual Cards – A virtual card has a status of active (virtual) and a working balance of $0.00.

Card Type	Card Status	Working Balance
Customer Valued Card	Active	$0.00
Pre-Valued Card	Inactive	$0.01 or Greater
Virtual Card	Active	$0.00

Interacting with Your SVS Cards (Request/Response)

It is essential to know the SVS Card Type with which you will be interacting. The SVS Card Type will determine the transaction set you will use to interact with your SVS Card.

All successful SVS Transactions consist of a Request, generated by you the Client, and a Response, generated by SVS.

For every one (1) request submitted by the Client, SVS will return one (1) response. Any response from SVS, not just an approval, constitutes a successful transaction. In the special circumstance where the Client submits a request and receives no response from SVS, it is the Client's responsibility to maintain the integrity of the SVS Transactional Database. Refer to Dup-check and Reversal processing style for further information.

Request Types

In this section we will discuss some of the most common types of requests. There are three different categories of requests we will discuss in this section:

- Credit Requests
- Debit Requests
- Non-Financial Requests

Credit Request

There are two types of Credit Requests, Initial and Secondary.

- Initial Credit Requests
  - Issue[1] – The initial credit transaction performed on a customer valued card.
  - Activate[1] – The initial credit transaction performed on a pre-valued card.
  - Issue Virtual Card[1] – The initial credit transaction performed on a virtual card.
  - Merchandise Return – Can be used as the initial credit transaction on a customer valued card.
  - Enable Card[1] - The initial credit transaction performed on a pre-valued card

After the initial credit request has been successfully completed, then all three types of SVS Cards are available for Secondary Credit Requests.

- Secondary Credit Requests
  - Recharge – A credit transaction that increases the working balance of your SVS Card.
  - Merchandise Return – Can be used as a secondary credit transaction to increase the working balance of your SVS card.

[1] The Issue, Activate, and Issue Virtual Card transactions are one-time transactions. These transactions may only be performed once in the lifetime of your SVS Card.

Debit Requests

A debit request is defined as a request to decrease the working balance of a SVS Card. This can be accomplished in either a one-step transaction or a two-step transaction.

- One-Step Debit Requests

  A one-step debit request consists of a single transaction to decrease the working balance of your SVS Card.

  - Redemption – The most common type of one-step debit request is a redemption request. The requested amount is deducted from the working balance of your SVS Card. When the requested amount exceeds the available balance, your SVS BIN range can be set up to return one of two things:
    - Insufficient Funds[2] – In this scenario SVS will return an insufficient fund response and there will be no effect to the working balance of your SVS Card.
    - Split-Tender – In this scenario SVS will return an approval response along with the amount approved. The amount approved has been deducted from the working balance of your SVS Card.
  - Cash Back – A one-step debit request that deducts the entire available balance amount from the working balance of your SVS Card. The amount deducted from your SVS Card will be returned in the response.
  - Tip – A one-step debit request that is similar to the redemption request. The only difference with the tip request occurs during the reporting that SVS provides.
- Two-Step Debit Requests (Pre-Authorization & Pre-Authorization Complete)

  A two-step debit request consists of a series of two transactions to ultimately decrease the working balance of your SVS Card. This two-step process is commonly implemented for fuel pump and website checkouts.

  - Pre-Authorization – The first step of a two-step debit request. The pre-authorization request increases the "out auth" amount of your SVS Card up to the amount of the working balance.
  - Pre-Authorization Complete – The second step of a two-step debit request. This request will decrease the working balance of your SVS Card by the requested amount. Also, the full amount of the associated pre-authorization request is released from the outstanding "out auth" amount of your SVS Card.
  - Pre-Authorization Clear ($0.00 Complete) – This is another option for the second step of a two-step debit request. This request will release the full amount of the referenced pre-authorization transaction. This is accomplished by sending in a pre-authorization complete request with an amount of $0.00, referencing the original pre-authorization.

[2] This type of setup will limit the reports that SVS can provide to you as a client. A business decision needs to be made on which setup SVS will apply to your BIN range.

Non-Financial Request

A non-financial request is defined as a transaction that does not directly affect the working balance of a SVS Card. First we will discuss three non-financial requests that will not affect the working balance of your SVS Card.

- Balance Inquiry – A non-financial request that will return the available balance of your SVS card.
- Enhanced Balance Inquiry – A non-financial request that will return the card balance and the last 25 financial transactions performed against the card.
- Network – An echo test message which can be used to confirm SVS web services are available.

There are two transactions that can indirectly alter the working balance of your SVS Card. The indirect alteration occurs because these requests undo a previously approved request.

- Cancel – A non-financial request that cancels the referenced transaction. This transaction is used when a user wants to cancel a previous transaction.
- Reversal – A non-financial request that reverses the referenced transaction. This request should only be generated automatically by the system when a failed transaction occurs.

Response Types

As stated earlier, for every one request received by SVS, a response will be returned to the client. In this response message, SVS will send a return code and a description, in English. This return code will reflect the result of the request.

The return code will contain a value whether or not the request was approved or denied and, if denied, the reason for denial. The Client or point-of-sale integrator may maintain a return-code mapping table, so a readable interpretation in the localized language can be provided to an individual customer at the merchant’s user interface.

It is noted that not all return codes are applicable to all request types. All return codes, aside from “01” and “15”, should display failure to the end user. A response code of “15” should set in motion the failed message logic of your program. After three corrective attempts (dup-check or reversal) with a response code of “15” your application should display failure to the end-user.

Return code	Return description
01	Approval
02	Inactive Card
03	Invalid Card Number
04	Invalid Transaction Code
05	Insufficient Funds
06	No Previous Authorizations
07	Invalid Message
08	No Card Found
09	Insufficient Funds due to Outstanding Pre-Authorization.
10	Denial - No Previous Authorization
11	No Authorization Number
12	Invalid Authorization Number
13	Maximum Single Recharge Exceeded
14	Maximum Working Balance Exceeded
15	Host Unavailable
16	Invalid Card Status
17	Unknown Dealer/Store Code – Special Edit
18	Maximum Number of Recharges Exceeded
19	Invalid Card Verification Value (CVV) or Secondary Security Code (SSC)
20	Invalid PIN Number or PIN Locked
21	Card Already Issued

Return code	Return description
22	Card Not Issued
23	Card Already Used
24	Manual Transaction Not Allowed
25	Mag Stripe Read not Valid
26	Transaction Type Unknown
27	Invalid Tender Type
28	Invalid Customer Type
29	*n/a*
30	Max number of redemption exceeded
31	Invalid currency code
32	Invalid Server ID (restaurant only – server ID code is invalid)
33	Frozen card or Unknown
34	Invalid Amount (transaction amount does not match the pre-valued card dollar amount)
35	*n/a*
36	Transaction rejected – invalid transaction promotion-wide or invalid for originating merchant and store. (DPS)
37	Transaction rejected – invalid merchant and store combination for promotion. (DPS)
38	Transaction rejected – exceeded maximum number of promotion cards allowed for a single POS transaction. (DPS fraud control)
42	Transaction declined – transaction amount less than the required minimum. (DPS)
98	Timeout/Error Communicating with third party authorizer
99	Failure in retrieving data (Enhance Balance Inquiry Only)

SVS Gift Card Web Services API

The SVS Web Services is based on open standards[3] known collectively as "Web Services", which include the following components:

- Simple Object Access Protocol (SOAP v1.1)

  An XML based protocol that consists of three parts:

  – An envelope that defines a framework for message content description and message processing.

  – A set of encoding rules for expressing instances of application-defined datatypes.

  – A convention for representing remote procedure calls and responses.

  For further information about the SOAP v1.1 Protocol visit:

  http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

  Web Services Description Language (WSDL v 1.1)

  As communications protocols and message formats are standardized in the web community, it becomes increasingly possible and important to be able to describe the communications in some structured way. WSDL addresses this need by defining an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages.

  For further information about the WSDL Protocol visit:
  http://www.w3.org/TR/wsdl

  The most current version WSDL file for the SVS Web Services is located at:

  Test:

  https://webservices-cert.storedvalue.com/svsxml/v1/services/SVSXMLWay/wsdl/SVSXMLWay.wsdlProduction:

  https://webservices.storedvalue.com/svsxml/v1/services/SVSXMLWay/wsdl/SVSXMLWay.wsdl

  Important Networking Consideration:

  Please note that SVS runs SVSXML services in multiple datacenters to handle failover and redundancy. The URL above (https://webservices.storedvalue.com) will resolve to the following range of IPs depending on which data center the services are running out of.

  If you have firewall rules in your organization that would affect SVSXML service calls, please open up the following IPs ranges to support both of our data centers.

  65.82.130.0/24 - Louisville

  65.82.131.0/24 - Brentwood

  Note: Applications that interact with SVS Web Services should be constructed utilizing the above WSDL file.

- XML Schema Definition Language (XSD)

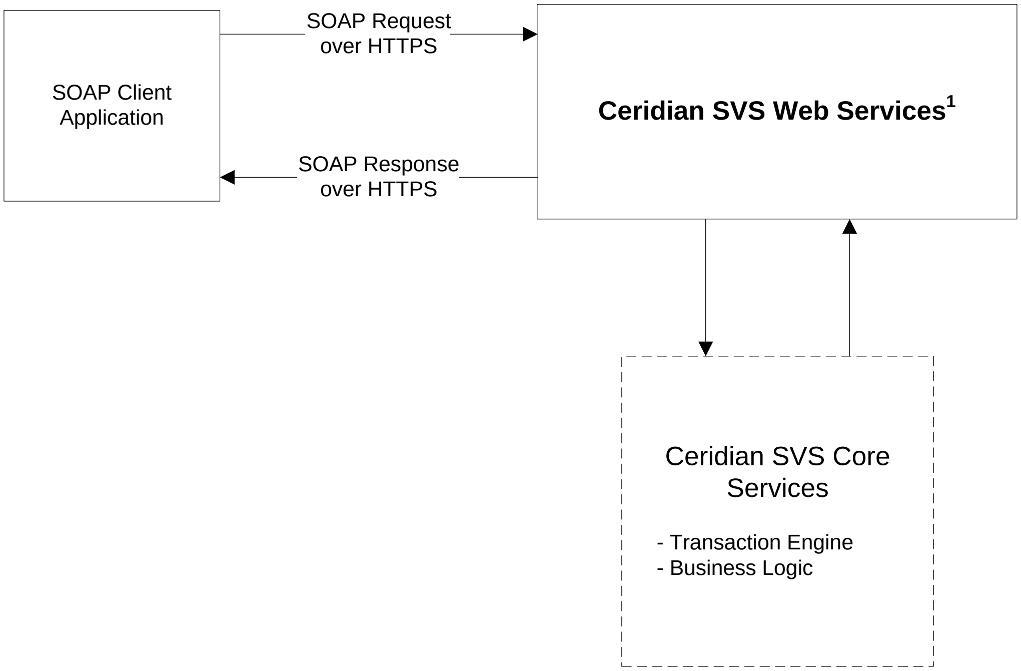
[3] These standards are supported by a wide variety of development tools across multiple platforms.

An XML Language for describing and constraining the content of XML documents.

For more information about the XSD Protocol, visit: http://www.w3.org/TR/xmlschema-1/

Stored Value Solutions Web Services Architecture

SVS Web Services are a combination of client-side and server-side schemas in addition to hardware, software, and core services.



In an object-oriented processing model, the interface to SOAP requests/responses is an object in your application's native programming language. Your third-party SOAP client generates business-object interfaces and network stubs ***from the SVS provided WSDL file***, which specifies the SVS SOAP message structure, its contents, and the SVS API service bindings. A business application works with data, in the form of object properties, to send and receive data by calling object methods. The SOAP client handles the details of building the SOAP request, sending it to the SVS service, and converting the response back to an object.

SVSXMLv1 application includes existing transaction and the following enhancements

- New Enable Card transaction addition
- Existing transaction Redemption and Preauthorization includes sku element in both request and response
- Existing transaction Issue Gift card incudes incentive number element in the response
- New Issue Virtual Third Party transaction addition

[1]Please note the new SVSXMLv1 *URLs listed below for SVS **Test** and **Production** Web Services:*

Test:
https://webservices-cert.storedvalue.com/svsxml/v1/services/SVSXMLWay

Production

https://webservices.storedvalue.com/svsxml/v1/services/SVSXMLWay

Security

The SVS Web Services API is protected to ensure that only authorized SVS customers may use the services. There are three primary levels of security:

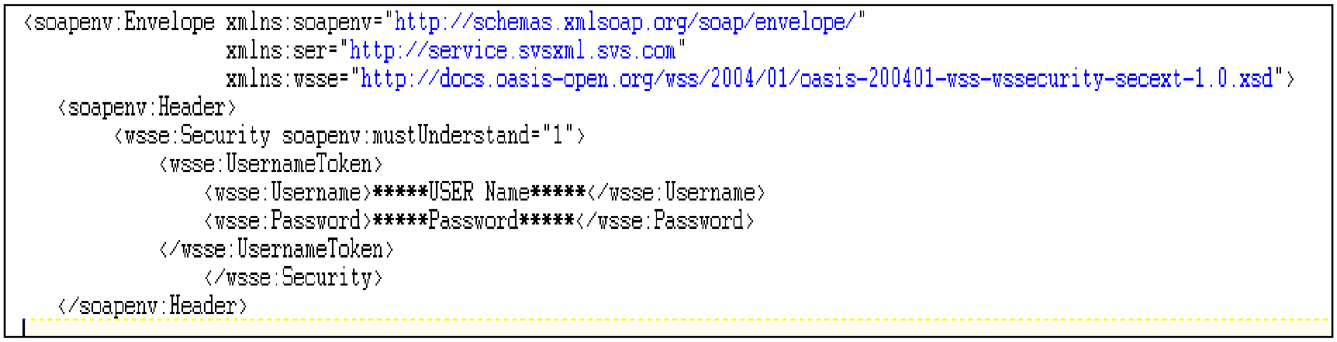
- Basic User Authentication (username and password)
- Public key encryption. (SVS Web Services uses a Verisign© public 2048-bit certificate)
- Secure Socket Layer (SSL) data transport

A failure in authentication at any one of these levels will deny access to the SVS Web Services API.

Basic User Authentication (username and password)

For the security of your business, SVS must authenticate that merchants are permitted to initiate a transaction before they make one. SVS will authenticate all requests. If a failure occurs during authentication, a SOAP security fault is returned.

In the SOAP request header, your SOAP client must set the Username and Password elements with the information provided by SVS. The following is the header of a SOAP request:



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ser="http://service.svsxml.svs.com"
                  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
   <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1">
            <wsse:UsernameToken>
                <wsse:Username>*****USER Name*****</wsse:Username>
                <wsse:Password>*****Password*****</wsse:Password>
            </wsse:UsernameToken>
                </wsse:Security>
   </soapenv:Header>
```

A couple things to note about the above example:

- Line 3 dictates the format and version of the UsernameToken that your SOAP Client will generate.

  xmlns:wsse=http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

  This namespace must be as above, to generate a correctly formatted UsernameToken that will pass authentication by the SVS Web Services.

- Lines 7 and 8 are the elements that contain the username and password. These values must be generated by your SOAP Client as plain text. <u>Encrypting these values will cause a failure of authentication</u> by the SVS Web Services.

  Note: You should obtain your unique username and password prior to your testing start date by requesting them from your SVS account or implementation manager.

Public Key Encryption

Production SVS Web Services uses a Verisign© public 2048-bit certificate.

Test SVS Web Services uses a public 2048-bit self-signed certificate.

SSL Data Transport

All data must be transported over the Secure Hyper Text Transport Protocol (also known as SHTTP or HTTPS), which relies on the Secure Sockets Layer (SSL) data communications protocol.

Processing Styles

SVS Web Services supports two separate processing styles. An SVS representative will guide you on which processing style you should utilize.

- Dup-Check is most commonly used for fuel-pump and website checkouts. Reversal is most commonly used for existing clients because it allows for maximum code reuse.

There are two primary differences between these processing styles.

- Cancels, Reversals, Network Messages and swiped transactions are available with the Reversal Process. They are not allowed with Dup-Check.

Failed messaging is handled differently between the two styles. Further information about failed messaging is listed below

The failed message logic is required. You must implement either dup-check or reversal logic as per the specifications listed below

The value you provide in the element **<checkForDuplicate>** dictates which process you will follow. This element is of type Boolean and must be provided in all requests to the SVS Gift Card Web Services (except Cancel, Reversal, and Network messages).

Dup-Check Process

The first process we will discuss is the Dup-Check process. Sometimes messages can get lost or will fail processing due to errors. SVS has implemented duplicate detection (dup-check) based on the alpha-numeric element <transactionID>. Usually, reversal logic has been deployed within the merchant's POS platform, but the dup-check process that has been implemented by SVS eliminates the need to develop a similar reversal process within the e-commerce environment.

For every incoming request, the client should transmit a unique transactionID. SVS stores the values of the request in the database. SVS will process the transaction request, store the result in the database, and send the result in a response message to the merchant.

If the merchant does not receive a response within a set timeframe, the merchant can resend the request. If SVS received the transaction, as identified by the transaction ID, it will send the response that is stored in the database as if the transaction were approved. If the transactionID is matched but the result wasn't available or the result was a denial, SVS will try to reprocess the transaction, store the result, and send the result in a response message to the merchant.

If there is not a match in the database for this transactionID, SVS will process the request as a new transaction processing request.

All records used for duplicate detection will be retained in the transaction processing environment for ninety (90) days. This implies that a transactionID is unique for the ninety (90) days that those records are maintained. It should be noted that the integrity of the transactionID is the responsibility of the merchant. If a transactionID recurs within the period of ninety (90) days and does not relate to a retry of a transaction request, unexpected results may occur.

NOTE: TransactionID is an alpha-numeric field

- Description of Key Dup-Check Elements

  ```
  <checkForDuplicate>true</checkForDuplicate>
  </stan>
  <transactionID>2143650000000001</transactionID>
  ```

  checkForDuplicate- When choosing to implement the Dup-check process, the checkForDuplicate element will contain the boolean value of "true" or "1" for all requests.

  transactionID- The element transactionID must be a unique identifier of your original transaction for 90 days. The first six (6) elements are the merchant number inversed in pairs of two (2). In the example above, merchant number 123456 would translate to 214365 as the first six (6) digits of the transactionID. You will utilize the remaining ten (10) digits to ensure a unique transactionID for a period of no less than 90 days.

  stan- Please note that the element "<stan>" is not utilized when using the Dup-Check process.

- Dup-Check Failed Messaging

  A transaction should be considered failed when one of the two following events occur:

  – SOAP Client generates a request but no reply is received from the SVS Host

  – SOAP Client generates a request and a response of 15 is received from the SVS Host

  When a failed transaction occurs, it is the client's responsibility to maintain the integrity of the SVS Transactional Database. When you are utilizing the Dup-Check Process, the following logic can be utilized

```
 1  execute web services call
 2
 3  SET count = 0
 4
 5  WHILE (response = 15 or No Response within 10 seconds) AND (count < 3)
 6       Retransmit an exact duplication of the request to  SVS Host
 7       INCREMENT count
 8  ENDWHILE
 9
10  IF count = 3 THEN
11       mark transaction for manual follow-up
12       end of transaction
13  ELSE
14       continue with normal processing
15  ENDIF
```

LINE 5 – SVS recommends that a failed transaction be retransmitted three times before displaying a failure to the end user.

LINE 6 – When a failed transaction occurs using the Dup-Check Process, an exact transaction should be sent to the SVS Host. It is very important that the same transactionID be sent for the retry of the request.

LINE 11 – After 3 attempts of the transaction, there is still a possibility that the request was approved but your SOAP Client did not get the response. In this case, the transaction should be flagged for manual follow-up to ensure the integrity of the SVS Transactional Database is upheld.

Reversal Process

The next process that we will discuss is the Reversal process. SVS has developed the Reversal Process with our existing clients in mind. This process will allow for maximum code reuse due to the fact that the business logic of this process is almost identical to the previous API offerings. An example of the key elements utilized is listed below:

- Description of Key Elements

```
<checkForDuplicate>false</checkForDuplicate>
<stan>112233</stan>
</transactionID>
```

  - checkForDuplicate- When choosing to implement the Reversal process, the checkForDuplicate element will contain the Boolean value of "false" or "0" for all requests.
  - transactionID- Please note that the element "transactionID" is not utilized when using the Reversal process.
  - stan- Systems Trace Audit Number. STAN is a unique sequence number assigned to a transaction at its inception. The STAN remains unchanged throughout the life of the transaction. Important notes regarding STAN-generation:
    - STAN is a 6-digit numeric field.
    - The sequence number should be unique ***to the card*** for every transaction type, including balance inquiries.
    - Sequence number can be incremented from 1-999999; however, the count should NOT be reset to 1 at the beginning of each day or upon power being cycled on a device.
    - Duplicate STAN numbers:
    - It is possible to have duplicate sequence numbers occur in the same day across the enterprise, depending on transaction load and whether sequence numbers are generated locally at the POS from a store controller or from the client's host. This is generally not problematic unless duplicate numbers occur on the same card.
    - The STAN is stored as the transaction's sequence number in the SVS Transaction History.

- Failed Reversal Transaction Process

  A transaction should be considered failed when one of the two following events occur:

  – SOAP Client generates a request but no reply is received from the SVS Host

  – SOAP Client generates a request and a response of 15 is received from the SVS Host.

  When a failed transaction occurs, it is the client's responsibility to maintain the integrity of the SVS Transactional Database. When you are utilizing the Reversal Process, the following logic can be utilized.

```
 1  execute web services call
 2
 3  SET count = 0
 4
 5  WHILE (response = 15 or 98 or No Response within 10 seconds) AND
    (count < 3)
 6       Transmit Reversal Request referencing original transaction
 7       INCREMENT count
 8  ENDWHILE
 9
10  IF count = 3 THEN
11       mark transaction for manual follow-up
12       End of transaction
13  ELSE
14       continue with normal processing
15  ENDIF
```

LINE 5 – SVS recommends that a failed transaction be retransmitted three times before displaying a failure to the end user.

LINE 6 – When a failed transaction occurs using the Reversal Process, a reversal request is submitted which references the original transaction. This reference is done by using the same value in the stan, cardNumber, and transactionAmount elements.

LINE 11 – After three (3) attempts of the transaction, there is still a possibility that the request was approved but your SOAP Client did not get the response. In this case, the transaction should be flagged for manual follow-up to ensure the integrity of the SVS Transactional Database is upheld. For a diagram of this process, please refer to Appendix C: Diagrams.

Supported Transactions

The transaction sets for both the Dup-Check and Reversal processes are very similar. The following table lists the full transaction offerings for both processing styles.

Transaction Type	Request Type	Dup-Check	Reversal
Issue	IssueGiftCardRequest	Supported	Supported
Activate	CardActivationRequest	Supported	Supported
Issue Virtual Card	IssueVirtualGiftCardRequest	Supported	Supported
Merchandise Return	MerchandiseReturnRequest	Supported	Supported
Reload	CardRechargeRequest	Supported	Supported
Redemption	RedemptionRequest	Supported	Supported
Cash Out	CashBackRequest	Supported	Supported
Tip	TipRequest	Supported	Supported
Pre-Authorization	PreAuthRequest	Supported	Supported
Pre-Authorization Complete	PreAuthCompleteRequest	Supported	Supported
Balance Inquiry	BalanceInquiryRequest	Supported	Supported
Cancel	CancelRequest	Not Supported	Supported
Reversal	ReversalRequest	Not Supported	Supported
Network	NetworkRequest	Not Supported	Supported
Enhanced Balance Inquiry	EnhancedBalanceInquiryRequest	Supported	Supported
Enable Card	EnableCardRequest	Supported	Supported
Issue Virtual Third Party	IssueVirtualThirdPartyRequest	Supported	Supported

Appendix A: SOAP Envelope Example

Transaction processing with SVS Web Services begins with the gift card client submitting a SOAP transaction request message to the appropriate SVS Web Services endpoint. The SVS Web Services then processes the transaction request and returns a SOAP transaction response message which must be interpreted by the gift card client to determine if the transaction request was approved, declined, or if message failure has occurred. Examples of a SOAP transaction request and a SOAP transaction response follow.

SOAP Transaction Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ser="http://service.svsxml.svs.com"
                  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">
   <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1">
            <wsse:UsernameToken>
                <wsse:Username>*****USER Name*****</wsse:Username>
                <wsse:Password>*****Password*****</wsse:Password>
            </wsse:UsernameToken>
                </wsse:Security>
   </soapenv:Header>

   <soapenv:Body>
      <ser:balanceInquiry>
         <request>
            <amount>
               <amount>0</amount>
               <currency>USD</currency>
            </amount>
            <card>
               <cardCurrency>840</cardCurrency>
               <cardNumber>6006499911069901970</cardNumber>
               <pinNumber>1686</pinNumber>
               <cardExpiration>?</cardExpiration>
               <cardTrackOne>?</cardTrackOne>
               <cardTrackTwo>?</cardTrackTwo>            </card>
            <merchant>
               <merchantName>Test Bal</merchantName>
               <merchantNumber>69966</merchantNumber>
               <storeNumber>9999999999</storeNumber>
               <division>99999</division>
            </merchant>
            <invoiceNumber>?</invoiceNumber>
            <routingID>6006499911069901970</routingID>
            <stan>176149</stan>
            <transactionID>?</transactionID>
            <checkForDuplicate>false</checkForDuplicate>
            <date>2011-06-21T10:16:51</date>
         </request>
      </ser:balanceInquiry>
   </soapenv:Body>
</soapenv:Envelope>
```

SOAP Transaction Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Header/>
   <soapenv:Body>
      <p751:balanceInquiryResponse xmlns:p751="http://service.svsxml.svs.com">
         <balanceInquiryReturn>
            <authorizationCode>443112</authorizationCode>
            <balanceAmount>
               <amount>4431.12</amount>
               <currency>USD</currency>
            </balanceAmount>
            <card>
               <cardCurrency>840</cardCurrency>
               <cardNumber>6006499911069901970</cardNumber>
               <pinNumber>00001686</pinNumber>
               <cardExpiration xsi:nil="true"/>
               <cardTrackOne xsi:nil="true"/>
               <cardTrackTwo xsi:nil="true"/>
               <eovDate>20160101T0500Z</eovDate>
            </card>
            <conversionRate>1.000000</conversionRate>
            <returnCode>
               <returnCode>01</returnCode>
               <returnDescription>Approval</returnDescription>
            </returnCode>
            <stan>176149</stan>
            <transactionID>?</transactionID>
         </balanceInquiryReturn>
      </p751:balanceInquiryResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Appendix B: API Reference

Balance Inquiry

Used to verify the available funds on a card.

Note: Upon request, if Pre-authorizations are being utilized, the balance returned is the card's working balance less any outstanding pre-authorized holds on funds.

<BalanceInquiry**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<amount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	0.00
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2007-09-10T17:57:59 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020

<checkForDuplicate>		type="xsd:boolean"		true or false
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<BalanceInquiry**Response**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<couponCode>		type="xsd:string"	30	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

Enhanced Balance Inquiry

Used to verify the available funds on a card and retrieves a list of the previous 25 financial transactions on a specific card.

Note: Upon request, the balance along with the transaction history is returned for a specific card.

<EnhancedBalanceInquiry**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<amount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	0.00
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000

<EnhancedBalanceInquiry**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false
<locale>		type="xsd:string"		Required field (ex . en_US)

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

< EnhancedBalanceInquiry**Response**>

Element		Sub-Element	Data Type	Length	Example / Comments
<authorizationCode>			type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>			*type="impl:Amount"*		
		<amount>	type="xsd:double"	6	9999.99
		<currency>	type="xsd:string"	3	USD
<card>			*type="impl:Card"*		
		<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
		<cardNumber>	type="xsd:string"	19	6006491286999929112
		<pinNumber>	type="xsd:string"	8	08760234
		<cardExpiration>	type="xsd:string"	6	110908
		<cardTrackOne>[1]	type="xsd:string"		
		<cardTrackTwo>[1]	type="xsd:string"		
		<eovDate>	type="xsd:string"		20160101T0500Z
<conversionRate>			type="xsd:string"	8	Not utilized for this transaction
<stan>			type="xsd:string"	6	Not utilized for this transaction
<transactionID>			type="xsd:string"	16	Not utilized for this transaction

< EnhancedBalanceInquiry**Response**>					
Element		**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<TransactionHistList >	**<TransactionElement>**		type= "impl:ArrayOfTransactionElement"		TransHist rows
	<ApprovedAmount>		type="impl:Amount"		
		<amount>	type="xsd:double"		9999.99
		<Currency>	type="xsd:string"		USD
	<Date>		type="xsd:string"		Locale date ex. en_US(10/21/2008)
	<Time>		type="xsd:string"		HH:MM:SS
	<requestedAmount >		type="impl:Amount"		
		<amount>	type="xsd:double"		9999.99
		<Currency>	type="xsd:string"		USD
	<merchant>		*type="impl:Merchant"*		
		<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
		<merchantNumber>	type="xsd:string"	6	061286
		<storeNumber>	type="xsd:string"	10	0000009999
		<division>	type="xsd:string"	5	00000
	<eovDate>[2]		Type="xsd:string"		20160101T0500Z
	<trnsAmount>		type="impl:Amount"		
		<amount>	type="xsd:double"		9999.99
		<Currency>	type="xsd:string"		USD

< EnhancedBalanceInquiry**Response**>					
Element		**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
	<transactionType>		type="impl:TransactionType"		
		<trnsCDindicator>	type="xsd:string"		Credit/Debit Indicator (C/D)
		<trnsCode>	type="xsd:string"		Transaction Code
		<trnsDescription>	type="xsd:string"		Type of Transaction
	<decimalPlaces>		type="xsd:string"		2
<returnCode>			type="impl:ReturnCode"		
		<returnCode>	type="xsd:string"	2	01
		<returnDescription>	type="xsd:string"		Approved
<preAuthAmount>			*type="impl:Amount"*		
	<amount>		type="xsd:double"	6	9999.99
	<currency>		type="xsd:string"	3	USD
<invoiceNumber>			type="xsd:string"	8	00000001

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

[2] **eovDate in Card represents current eov date per card. eovDate in transaction element is the eov date at the time of that given transaction.**

Card Activation Message Layout

Used to activate a previously inactive pre-valued card. Activating a card makes the funds associated with the card available to the consumer.

Note: This message type is used for **pre-valued** cards only.

SVSXML v2 includes coupon code element in request to support Coupon code processing

<cardActivation**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<activationAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS))
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	8612060000000001
<checkForDuplicate>		type="xsd:boolean"		true or false
<campaignCode>		type="xsd:string"	20	

<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO
[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing				

<cardActivation**Response**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456

<cardActivation**Response**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<transactionID>		type="xsd:string"	16	8612060000000001
<couponCode>		type="xsd:string"	30	NEWYEAR2017
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

Cancellation Message Layout

Used to manually void or post void a previous transaction. It is possible to cancel a transaction until it is reconciled or reversed. **Note:** In order to cancel funds held by a pre-authorization, send a zero dollar ($0) Pre-Authorization Completion transaction.

Note: On a timeout, resend the Cancel transaction up to 3 times, with a delay between each attempt, until a response is received to the cancellation.

Note: Cancel transactions can be sent for any original transaction except Cancels, Reversals, and Pre-Authorizations.

Note: Cancel transactions are **not available when utilizing Dup-Check logic**. Please note <checkForDuplicate> is not present in this request.

SVSXML v2 includes coupon code element in request to support Coupon code processing

<Cancel**Request**>	(Not available for Dup-Check Process)			
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<campaignCode>		type=”xsd:string”	20	

<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<CancelResponse>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	*type="xsd:string"*		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<campaignCode>		type="xsd:string"	20	
<couponCode>		type="xsd:string"	30	NEWYEAR2017
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

Card Recharge Message Layout

Used to add funds to a card that has already been issued. A Recharge cannot be the first transaction on a card.

<CardRecharge**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<rechargeAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	30	NEWYEAR2017

<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<CardRechargeResponse>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	type="xsd:string"		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<campaignCode>		type="xsd:string"	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<CardRechargeResponse>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**

Cash Back Message Layout

Used to debit a card for the total available balance of the card, leaving a zero dollar ($0) balance.

Note: Clients should check with their legal department concerning state laws governing the ability to offer cash back when a card’s balance reaches a specific amount.

<CashBack**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<cashBackAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)

<CashBack**Request**>

Element	Sub-Element	Data Type	Length	Example / Comments
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<CashBack**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<eovDate>	Type=”xsd:string”		20160101T0500Z
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

Issue Gift Card Message Layout

Used to place value on a customer valued card. The Issue Gift Card transaction must be the first transaction on a card. The amount requested is the amount of the issue.

SVSXML v1 includes incentive element in response to support Promotional code processing.

<IssueGiftCard**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<issueAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false

<IssueGiftCard**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<campaignCode>		type="xsd:string"	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<IssueGiftCard**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<incentiveNumber>		type="xsd:string"	24	0000000000001IncentiveCod
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<IssueGiftCard**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**

Issue Virtual Gift Card Message Layout

Generates a card number and PIN (with no physical card produced) from a pool of card numbers specifically ordered as virtual cards.

<IssueVirtualGiftCard**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<issueAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false
<campaignCode>		type="xsd:string"	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

<IssueVirtualGiftCard**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**

<IssueVirtualGiftCard**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<IssueVirtualGiftCard**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments

Merchandise Return Message Layout

Used to electronically apply value to a card from returned merchandise. A Merchandise Return transaction can be the first transaction on a new card or it can be applied to a previously used card.

<MerchandiseReturn**Request**>

Element	Sub-Element	Data Type	Length	Example / Comments
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<returnAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020

<checkForDuplicate>		type="xsd:boolean"		true or false
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<MerchandiseReturn**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

Network Message Layout

An "echo" or "heartbeat" message can be used to validate the SVS authorization platform is available.

<Network**Request**>	(Not available for Dup-Check Process)			
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<networkCode>		type="xsd:string"	3	301
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)

<Network**Response**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<returnCode>		*type="impl:ReturnCode"*		
	<returnCode>	type="xsd:string"	2	00
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456

Pre-Authorization Message Layout

Used to place a reserve on funds on a card for a fixed period of time or until a Pre-Authorization Completion message is approved. This message type should always be followed by a Pre-Authorization Completion transaction.

SVSXML v1 includes sku element in both request and response to support Promotional cards that are only good for specific items.

For further information regarding the Pre-Authorization and Pre-Authorization Completion transactions please see Appendix C.

<PreAuth**Request**>

Element	Sub-Element	Data Type	Length	Example / Comments
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<requestedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false
<sku>		type="xsd:string"	84	IFirst SKU ISecond SKU IThird SKU

<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<PreAuth**Response**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<sku>		type="xsd:string"	84	IFirst SKU ISecond SKU IThird SKU
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

Pre-Authorization Completion Message Layout

Used to release the pre-authorized hold on funds and debit the card. For further information regarding the Pre-Authorization and Pre-Authorization Completion transactions, please see Appendix C.

<PreAuthComplete**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<transactionAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<PreAuthComplete**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

Redemption Message Layout

Used to debit a cash card when it is used to buy goods or services.

SVSXML v1 includes sku element in both request and response to support Promotional cards that are only good for specific items.SVSXML v2 includes coupon code element in request to support Coupon code processing

<Redemption**Request**>

Element	Sub-Element	Data Type	Length	Example / Comments
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<redemptionAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false
<sku>		type="xsd:string"	84	IFirst SKU ISecond SKU IThird SKU
<campaignCode>		type=”xsd:string”	20	

<couponCode>		type="xsd:string"	20	NEWYEAR2017

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<RedemptionResponse>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type="xsd:string"		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<sku>		type="xsd:string"	84	IFirst SKU ISecond SKU IThird SKU
<campaignCode>		type="xsd:string"	20	
<cardType>		Type="xsd:string"	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

Reversal Message Layout

Used to resolve situations arising from a POS timeout or network outage. Reversals are **not available when utilizing Dup-Check** logic. A reversal is not a cancellation. Reversal transactions can be sent for any original transaction except Reversals, Balance Inquiries, Pre-Authorizations, Cancels and Issue Virtual. The reversal transaction should be attempted until a response is received, with a maximum of three (3) attempts, at which time the transaction should be flagged for manual follow-up. Please note <checkForDuplicate> is not present in this request.

SVSXML v2 includes coupon code element in request to support Coupon code processing

<Reversal**Request**>	(Not available for Dup-Check Process)			
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		*type="impl:Card"*		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

<Reversal**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”	26	20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

Tip Message Layout

Used to debit the card for a gratuity

Note: The Tip transaction functions like a standard debit. Split-tender will be based on the client's preference.

<Tip**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<card>		type="*impl:Card*"		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<merchant>		type="*impl:Merchant*"		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<tipAmount>		type="*impl:Amount*"		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false

[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

<Tip**Response**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

Enable Card Message Layout

This transaction type is only used with the Dynamic Promotion Solution (DPS) product. Used during the issuance period to enable a promotional card. If the card is not enabled it will not be possible to redeem during the redemption period.

<EnableCardRequest>

Element	Sub-Element	Data Type	Length	Example / Comments
<amount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<eovDate>[2]	type=”xsd:string”		20160101T0500Z
	<cardCurrency>	type="xsd:string"		
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	84	08760234
	<CardExpiration>	type="xsd:string"	6	110908
	<CardTrackOne>[1]	type="xsd:string"		
	<CardTrackTwo>[1]	type="xsd:string"		
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000

<invoiceNumber>		type="xsd:string"	8	00000001
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<checkForDuplicate>		type="xsd:boolean"		true or false
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

[2] eovDate in Card represents current eov date per card. eovDate in transaction element is the eov date at the time of that given transaction.

<EnableCardReturn>

Element	Sub-Element	Data Type	Length	Example / Comments
<authorizationCode>		type="xsd:string"	6	
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<eovDate>[2]	type=”xsd:string”		20160101T0500Z
	<cardCurrency>	type="xsd:string"	3	840
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	84	08760234
	<CardExpiration>	type="xsd:string"	6	110908
	<CardTrackOne>[1]	type="xsd:string"		

	<CardTrackTwo>[1]	type="xsd:string"		
<conversionRate>		*type="impl:Merchant"*		
<returnCode>				
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approval
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<redeemWindow>		type="xsd:string"	28	2015010913022129991231235959
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)

[1] cardTrackOne and cardTrackTwo are not supported by dup-check style processing

[2] eovDate in Card represents current eov date per card. eovDate in transaction element is the eov date at the time of that given transaction.

Issue Virtual Third Party Message Layout

Generates a card number and PIN (with no physical card produced) from a pool of virtual inventory card numbers specifically ordered as virtual cards.

<IssueVirtualThirdParty**Request**>				
Element	**Sub-Element**	**Data Type**	**Length**	**Example / Comments**
<date>		type="xsd:string"		2011-08-15T10:16:51 (YYYY-MM-DDTHH:MM:SS)
<invoiceNumber>		type="xsd:string"	8	00000001
<issueAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<virtualInventoryID>		type="xsd:string"	6	123456
<merchant>		*type="impl:Merchant"*		
	<merchantName>	type="xsd:string"	22	Gift Card Merchant, INC
	<merchantNumber>	type="xsd:string"	6	061286
	<storeNumber>	type="xsd:string"	10	0000009999
	<division>	type="xsd:string"	5	00000
<routingID>		type="xsd:string"	19	6006491286999900000
<stan>		type="xsd:string"	6	123456 (HHMMSS)
<transactionID>		type="xsd:string"	16	2143651105080020
<checkForDuplicate>		type="xsd:boolean"		true or false
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

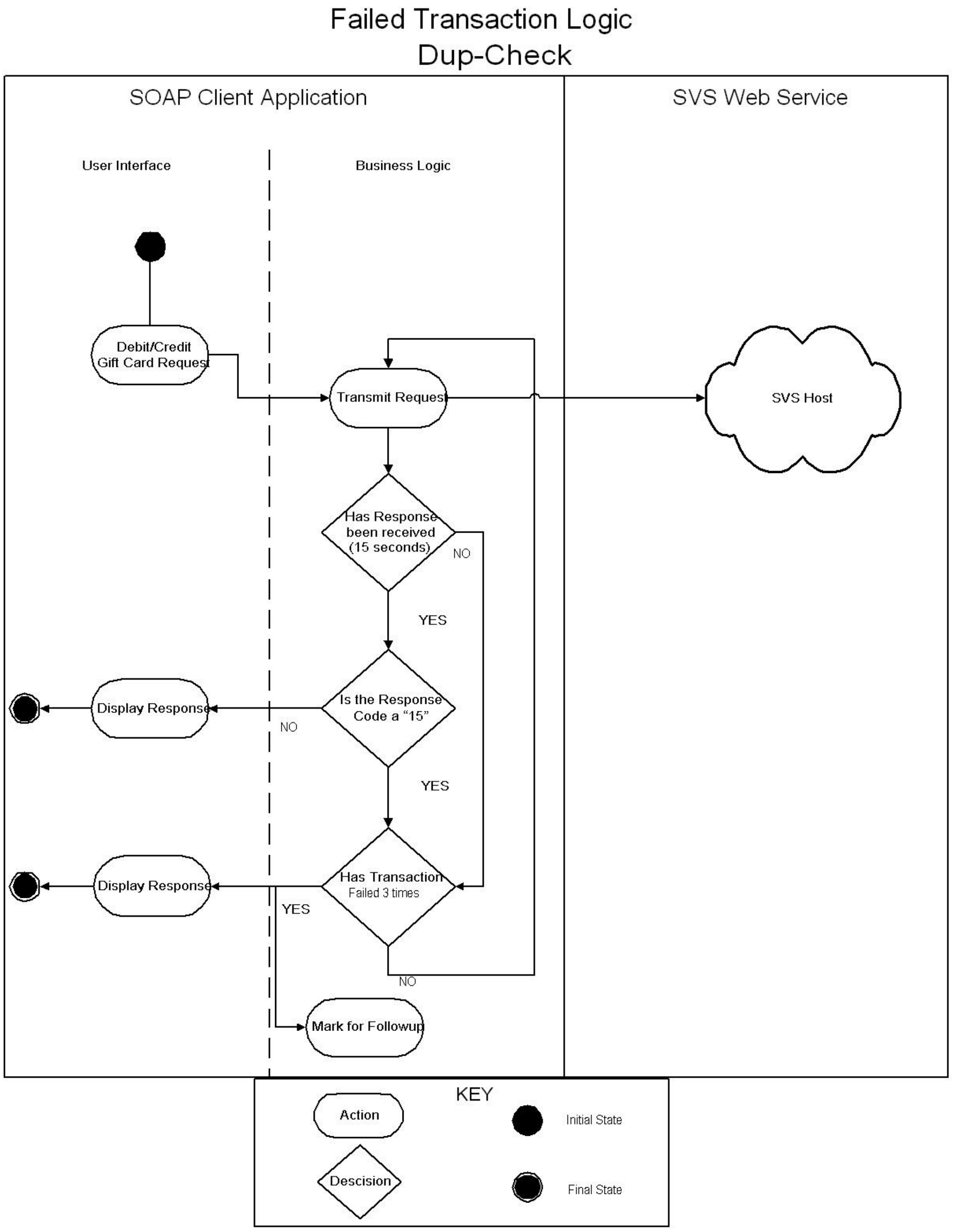
<IssueVirtualThirdParty**Response**>

Element	Sub-Element	Data Type	Length	Example / Comments
<approvedAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<authorizationCode>		type="xsd:string"	6	Not utilized for Gift Card Web Service
<balanceAmount>		*type="impl:Amount"*		
	<amount>	type="xsd:double"	6	9999.99
	<currency>	type="xsd:string"	3	USD
<card>		*type="impl:Card"*		
	<cardCurrency>	type="xsd:string"	3	840 (Numeric translation of USD)
	<cardNumber>	type="xsd:string"	19	6006491286999929112
	<pinNumber>	type="xsd:string"	8	08760234
	<cardExpiration>	type="xsd:string"	6	110908
	<cardTrackOne>[1]	type="xsd:string"		
	<cardTrackTwo>[1]	type="xsd:string"		
	<eovDate>	Type=”xsd:string”		20160101T0500Z
<conversionRate>		type="xsd:string"	8	1.000000
<returnCode>		type="impl:ReturnCode"		
	<returnCode>	type="xsd:string"	2	01
	<returnDescription>	type="xsd:string"		Approved
<stan>		type="xsd:string"	6	123456
<transactionID>		type="xsd:string"	16	8612060000000001
<campaignCode>		type=”xsd:string”	20	
<couponCode>		type="xsd:string"	20	NEWYEAR2017
<cardType>		Type=”xsd:string”	9	GIFT / MERCH / PROMO

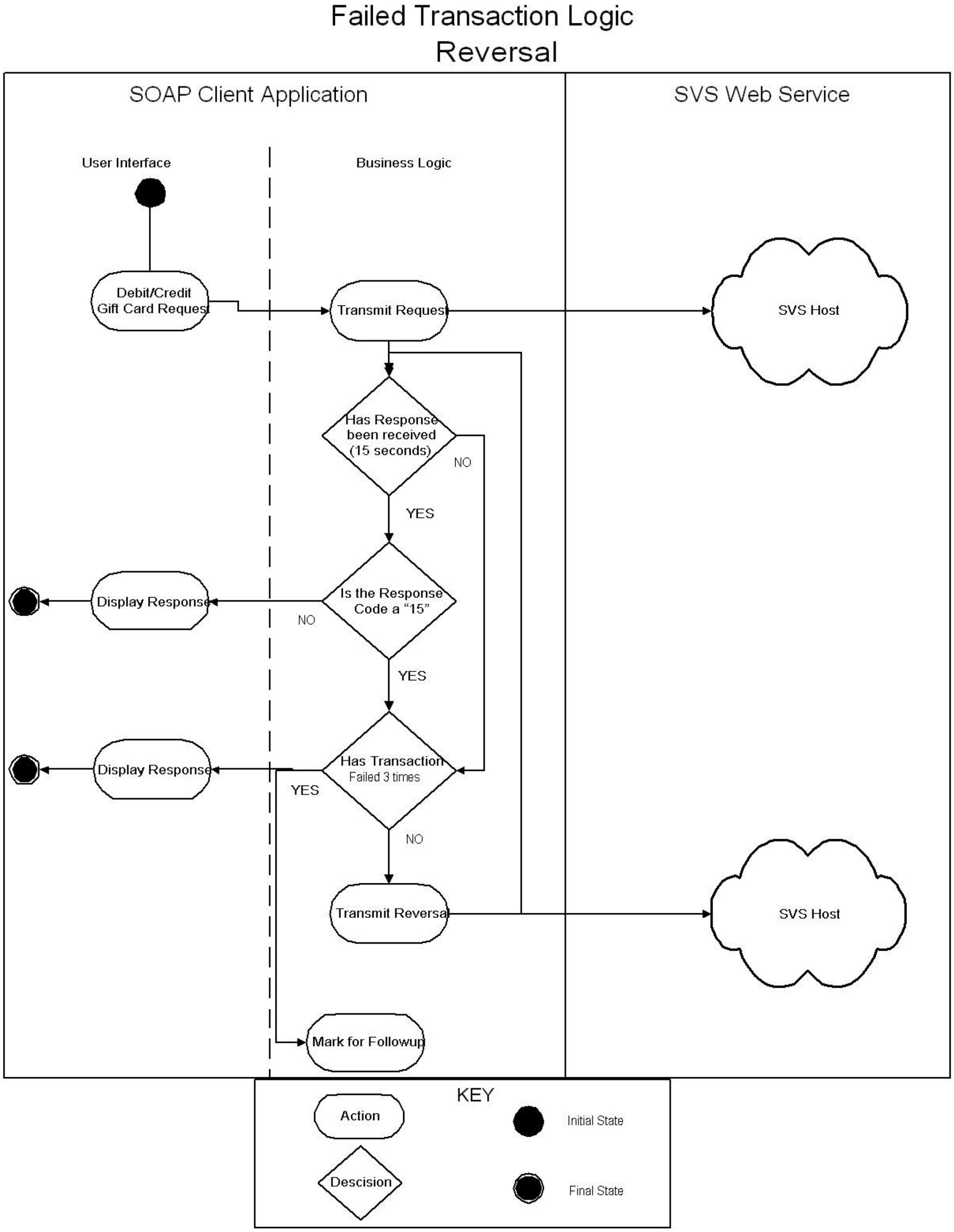
[1] **cardTrackOne and cardTrackTwo are not supported by dup-check style processing**

Appendix C: Diagrams

Failed Transaction Logic Dup-Check



Failed Transaction Logic Reversal

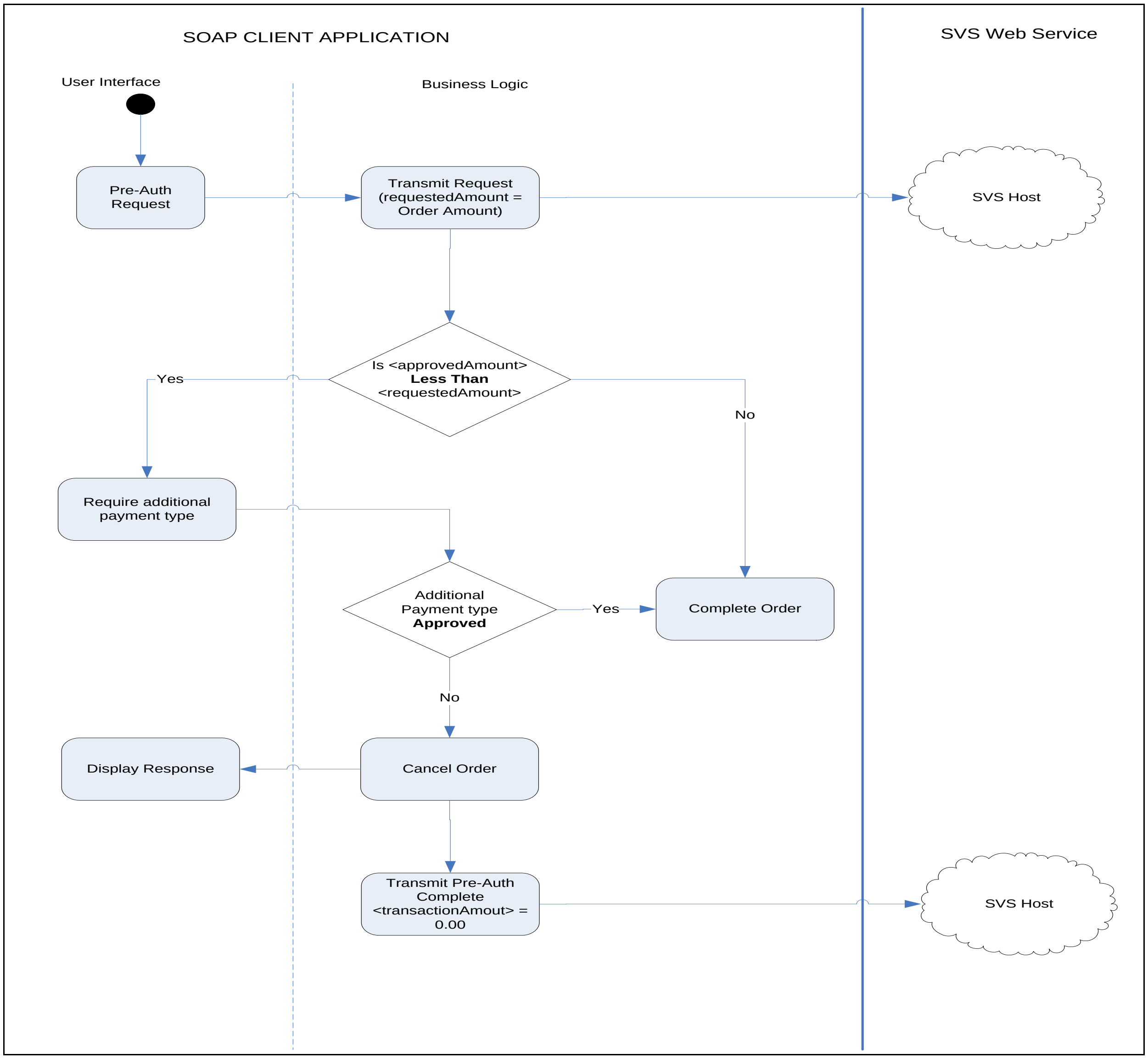


Pre-Authorization Transaction

The Pre-Authorization is used to place a reserve on the funds on a card until the Pre-Authorization Complete transaction is submitted to debit the card. Cancels or Reversals cannot be performed against this transaction.

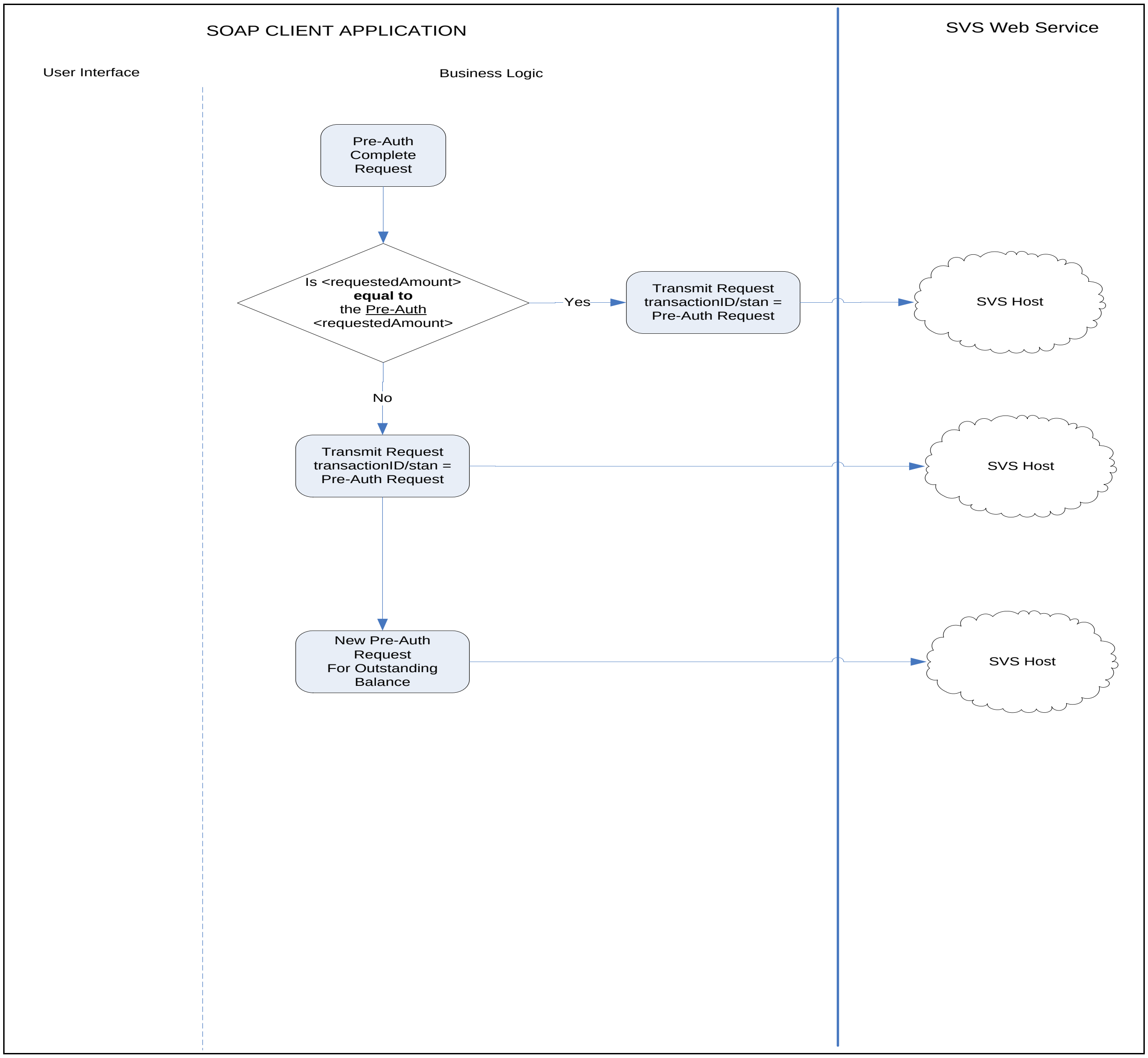
A Pre-Authorization hold is released in one of the following ways:

1. Pre-Authorization Completion request using the same **transactionID** (Dup-Check) or **stan** (Reversal Process) as the Pre-Authorization Request.
2. Batch Process – This process runs daily at a specified time to release Pre-Authorization (TRN190).
3. OSI – Corporate associated that have access to the OSI application can perform a card look up and release any outstanding pre-authorizations.


SOAP CLIENT APPLICATION
SVS Web Service
User Interface
Business Logic
Pre-Auth Request
Transmit Request (requestedAmount = Order Amount)
SVS Host
Is <approvedAmount> Less Than <requestedAmount>
Yes
No
Require additional payment type
Additional Payment type Approved
Yes
Complete Order
No
Display Response
Cancel Order
Transmit Pre-Auth Complete <transactionAmout> = 0.00
SVS Host

Pre-Authorization Completion Transaction

The Pre-Authorization Completion is the final step in completing a Pre-Authorization transaction. When the transaction is completed, the amount of the sale/order is sent to Stored Value Solutions in order to release the Pre-Authorization hold on funds and debit the account. The same **transactionID** or **stan** that was sent in the Pre-Authorization request must be used in the Pre_Authorization Completion message.


SOAP CLIENT APPLICATION
SVS Web Service
User Interface
Business Logic
Pre-Auth Complete Request
Is <requestedAmount> equal to the Pre-Auth <requestedAmount>
Yes
Transmit Request transactionID/stan = Pre-Auth Request
SVS Host
No
Transmit Request transactionID/stan = Pre-Auth Request
SVS Host
New Pre-Auth Request For Outstanding Balance
SVS Host

Appendix D: Transaction Glossary

Activate – initial transaction which changes the pre-valued gift card status to "active" within the SVS database, causing the pre-determined gift card balance to become available for use.

Balance Inquiry – transaction used to determine gift card balance

Cancel – transaction used to void the previous transaction performed on a gift card; balance inquiry and pre-authorization are not eligible for cancellation

Cashback – transaction used to decrease the gift card balance to $0

Enhanced Balance Inquiry – transaction used to determine gift card balance and displays a list of the previous 25 transactions on the card

Issue – initial transaction used to place value on a customer-valued gift card

Issue Virtual – initial transaction used to place value on a virtual gift card; virtual card number and PIN are selected from SVS virtual card inventory and returned on the SOAP response; physical plastic card does not exist

Merchandise Return – transaction used to add value to a gift card in exchange for returned merchandise and in lieu of a cash refund; may also be used as the initial transaction used to place value on a customer-valued gift card

Network – transaction used to confirm SVS web services are available; echo test

Pre-Authorization – transaction used to place a hold on all or part of the gift card balance until the preauthorization complete transaction occurs

Pre-Authorization Complete – transaction used to remove the pre-authorization hold on gift card funds and decrease the gift card balance

Recharge – transaction used to add value to a gift card

Redemption – transaction used to decrease the gift card balance

Reversal – system-generated void transaction submitted to SVS upon POS transaction timer expiration

Tip – transaction used to decrease the gift card balance

Enable Card - A transaction only used with the Dynamic Promotion Solution (DPS) product. Used during the issuance period to enable a promotional card. If the card is not enabled it will not be possible to redeem during the redemption period

Issue Virtual Third Party – initial transaction used to place value on a third party virtual gift card; virtual card number and PIN are selected from SVS virtual card inventory and returned on the SOAP response; physical plastic card does not exist

Appendix E: Message Element Glossary

amount – complex type composed of two sub elements

amount.amount – sub element of complex type amount which specifies the requested transaction amount

amount.currency – sub element of complex type amount which specifies the alphanumeric currency code representing local currency used at the point-of-sale; the gift card remaining balance reported in the SOAP response will also be denominated as POS local currency

campaignCode – A value sent by the POS/TILL system in Activation and Issue transactions that is then associated with the card. On redemption transactions the system will return the campaignCode and associated couponCode. Sending a reversal or cancel of an Issue/Activation will remove the campaignCode and its associated coupon code from the card account.

card – complex type composed of six sub elements

cardBalanceAmt - the card balance amount.

card.cardCurrency – sub element of complex type card present in the SOAP response which identifies the numeric currency code associated with the gift card

card.cardNumber – sub element of complex type card which specifies the gift card number associated with the transaction request

card.pinNumber – sub element of complex type card which specifies the gift card PIN and/or Secondary Security Code (SSC) associated with the transaction request. This field is divided into two subfields; the first 4 digits identify the secondary security code (SSC). The second subfield is 4 digits to be used identify the PIN.. **Each Sub-field should be right justified left zero filled**.

- **Secondary Security Code(SSC)** – Is a 4 digit code encoded into the barcode on the gift card. To be used when the card's barcode is scanned. Customers select the format of the barcode and sometime elect not to encode the secondary security code.
- **PIN** – Is a 4 digit code that is encoded on the gift card, typically under a scratch off. To be used when the end consumer is prompted to enter their PIN. Commonly used for ecommerce transactions or when the card number is manually keyed.

card.cardExpiration – sub element of complex type card which is not used in SVS web services processing

card.cardTrackOne – sub element of complex type card which specifies data encoded on the Track 1 magnetic stripe of the gift card; present in the SOAP request only when the POS terminal captures magnetic strip information and Track 2 data is not present

card.cardTrackTwo – sub element of complex type card which specifies data encoded on the Track 2 magnetic stripe of the gift card; present in the SOAP request only when the POS terminal captures magnetic strip information

card.eovDate – End of Validity date for that card at the moment of the response

cardType – The type of card being processed. This may have the values GIFT, MERCH, PROMO or some value either assigned at the POS or for a specific promotion. May be up to 9 characters in length.

checkForDuplicate – Boolean value present in the SOAP request which specifies the desired SVS web services processing style; DUP-CHECK processing is used when checkForDuplicate=true, REVERSAL processing when checkForDuplicate=false. Valid Boolean values are lower case true,false and 0 and 1

conversionRate – numeric value present in the SOAP response which specifies the conversion factor used to convert the transaction amount from the base currency of the gift card to the POS local currency specified in amount.amount of the SOAP request.

couponCode – A value sent by the POS/TILL system in Activation and Issue transactions that is then associated with the card. On redemption transactions the system will return the couponCode and mark it as used. Subsequent redemptions on the card will not return the coupon code. Sending a reversal or Cancel of a redemption will mark the coupon as unused. Sending a reversal or cancel of an Issue/Activation will remove the coupon code from the card account. This value is used in conjunction with the campaignCode.

currencySymbol –String value -The currency symbol of the amounts to displayed for a web balance/Transaction Amount

date – xsd:String value present within the SOAP request which specifies the date and time of transaction occurrence.

incentiveNumber - contains the promotion number that has been satisfied and the promotional code being returned. The value in the promotional code field can be used by the POS system as a COUPON or other trigger for further POS processing

invoiceNumber – client assigned data value present in the SOAP request which may be used to identify the terminal or POS where the transaction occurred; the invoiceNumber data value can represent an order number for online or catalog transactions.

locale – is used in the Enhanced balance Inquiry transaction to indentify the language to use when displaying the trnsDescription. The format is <language>_<country> ex. en_US.

merchant – complex type composed of four sub elements

merchant.merchantName – sub element of complex type merchant which specifies the name of the merchant submitting the gift card transaction

merchant.merchantNumber – sub element of complex type merchant which specifies the SVS assigned merchant number of the merchant submitting the gift card transaction

merchant.storeNumber – sub element of complex type merchant which specifies the client assigned store number of the merchant location submitting the gift card transaction

merchant.division – sub element of complex type merchant which specifies the client assigned division of the merchant location submitting the gift card transaction; SVS recommended practice is to supply store/division information in the ANI file rather than passing the division value within the SOAP request, the division sub element should be zero-filled for five digits when not in use.

posType – numeric value representing the method for consumer interaction in a transaction; values are 0-POS (traditional store),8-Website,27-Fuel Pump

preAuthAmount – pending amount being held against the card balance.

returnCode – complex type composed of two sub elements

returnCode.returnCode – sub element of complex type returnCode present in the SOAP response which specifies the approval or denial of a request submitted to SVS using a numeric value; see the response code listing within this document for further information

returnCode.returnDescription – sub element of complex type returnCode present in the SOAP response which provides descriptive text associated with the numeric returnCode received

routingID – a SVS assigned value which facilitates transaction routing within SVS transaction processing systems

stan – transaction identifying value used with SVS Reversal web services processing, must be unique at the gift card level; can be formatted as HHMMSS (HH = hour, MM = minute, SS = second) of transaction creation.

sku -an optional component of redemption and preauthourization processing that is used by the DPS product. SKU's may either be inclusive or exclusive depending about a flag in the data element. The host will validate the submitted SKU's and process the transaction as normal

virtualInventoryID– which specifies the SVS assigned merchant number of the third party merchant submitting the gift card transaction

TransactionHistList - the list of transactions for a particular card number when enhanced balance enquiry service is invoked.

TransactionHistList.TransactionElement – sub element of complex type EnhancedBalanceInquiryResponse which hold the details of a particular transaction row.

TransactionHistList. TransactionElement. approvedAmount – the amount approved on the transaction.

TransactionHistList.TransactionElement. currency – the currency code – ex USD used on the transaction.

TransactionHistList.TransactionElement.date – the date when the transaction occurred.

TransactionHistList.TransactionElement. decimalPlaces– is used to determine the decimal placement for the amount elements. The decimal placement is based on the type of currency used on the transaction.

TransactionHistList.TransactionElement.requestedAmount – the amount requested on the transaction and can differ from the approved amount.

TransactionHistList.Transaction Element. merchant – contains all elements of the merchant complex type

TransactionHistList.TransactionElement. TransactionType. trnsCDindicator – the Credit/Debit indicator

TransactionHistList.TransactionElement. TransactionType. trnsCode – the specific code assigned to the transaction.

TransactionHistList.TransactionElement. TransactionType. trnsDescription – the name(description) of the transaction which will be represented in the language indicated in the locale element.

TransactionHistList.TransactionElement.time – the time (HHMMSS) when the transaction took place.

TransactionHistList.TransactionElement.trnsAmount – the transaction amount.

TransactionHistList.eovDate – the End of Validity date for the card at the time of the specific transactions. (NOTE : Date can change from transaction to transaction)

transactionID – transaction identifying value used with SVS Dup-Check web services processing, must be unique for a period of 90 days; the first 6 digits of the transactionID are composed of the SVS assigned merchant number inversed in pairs of 2, for example merchant number 123456 becomes 563412; the remaining 10 digits can consist of a counter which increments sequentially from 0000000000 to FFFFFFFFFF before returning to 0000000000.