Oracle® Hospitality

Gift and Loyalty POS Web Services API
Release 8.5.1
E91264-04

September 2022

Contents

Preface

This document contains API integration information for Oracle Hospitality Gift and Loyalty.

Audience

This document is intended integrators to Oracle Hospitality Gift and Loyalty.

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received and any associated log files
- Screenshots of each step you take

Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at https://docs.oracle.com/en/industries/food-beverage/

Revision History

Date	Description of Change
October 2017	• Initial publication
September 2019	• Updated Attributes and Elements section
September 2022	• Updated Base Document – Transactional Request and Response Document with element definitions

1 iCare POS Interface Overview

This document specifies the Gift and Loyalty Stored Value, Loyalty and Promotions POS interface. This web interface may be used by POS solutions or other third parties to emulate POS transactions, performing real-time Stored Value (gift card, award cards, and credit or debit accounts), Loyalty and Promotion transactions, which include coupon validation and acceptance.

The POS interface (POSI) connects to the mymicros.net iCare OLTP Server (iOS) via SOAP over HTTPS1. iCare is the host that contains the transaction rules for payment, loyalty, and wards, records transactions, transaction details, and provides SOAP response messages to the POSI; iCare provides a rich transaction set including:

- Payment or Award Related Transactions & Inquiries:
  - Activate Payment Card(s)
  - Issue Payment Card(s)
  - Redeem or Finalize for Payment or Award Card(s)
  - Authorization for Payment or Award Card(s)
  - Reload Payment or Award Card(s)
  - Cash Out Payment or Award Card(s)
  - Transfer Payment, Award, or Loyalty Card Balance
  - Inquire Payment or Award Card Balance
- Loyalty Related Transactions & Inquiries:
  - Issue Loyalty Card Points and associated Awards
  - Inquire Loyalty Card Balance
  - Inquire Unique Items
- Promotion Related Transactions & Inquiries:
  - Accept Coupon
  - Inquire Coupon(s)
  - Issue Coupon

The iOS and POSI may support multiple and mixed currency, as well as mixed language operations.

Security Considerations

Store logon name and password are issued before a store can post web service call to the iOS, which can be changed per request. These credentials are highly confidential and should be stored in a secure place. It is the client's responsibility to maintain them. This information is used in SOAP request, see details at section "SOAP Request Format".

Connectivity between your Application and the iCare CRM Server Interface (CSI) is via SOAP over HTTPS preferably TLSv1.2.

Transaction Overview

The POSI posts a SOAP request to the iOS which processes it as required and returns a SOAP response. Typically the iOS requires 1-2 seconds to process the request before returning the response. Typical end-to-end transaction times using HTTPS over a broadband connection are 4-10 seconds. This includes the time required to perform the HTTPS authentication.*rintln(requestXml);*

```
CRC32 crc = new CRC32();

crc.update(requestXml.getBytes("UTF-8"));

String  crcValue = Long.toHexString(crc.getValue()).toUpperCase();

System.out.println(crcValue);
```

Optimally each store should connect via a persistent IP network, alternatively and as a backup a dial-up IP connection may be used. The interface does not require each store have a public and/or static IP address allowing connection via more economic third party ISP services. The interface is transport layer independent and as such supports many different IP connection methods such as DSL, frame, satellite, ISDN, cable and dial-up.

SOAP Request Format

This section describes the basic SOAP request and response formats.

This request is sent by the POSI to the iOS as an HTTPS "POST" request.

Each request includes four parameters:

Name	Description	Type	Example(s)
"in0"	SVC Message Document This parameter contains the request message; it is formatted as an XML document.	String	See later.
"in1"	Logon This is the store's name. This is used by the iOS to identify the source.	String(16)	"STORE"
"in2"	Password This is the store password. It is used by the iOS to validate the request and the source. The password is assigned arbitrarily and should be configured at the store.	String(16)	"Password"

“in3”	CRCValue This is the CRC32 hex value of “in0” and is used by the iOS to validate message integrity.	String	98A83356

The logon and password parameters must be coordinated with the iCare Data Center.

SOAP Response Format

Each response will include one parameter:

Name	Description	Type	Example(s)
“processRequestReturn”	SVC Message Document This parameter contains the response message; it is formatted as an XML document.	String	See later.

The SOAP “wsdl” document is shown in Appendix II. A SOAP trace showing illustrating a typical request and response pair is shown in Appendix III.

2 SVC Message Format

Each SVC message is sent as an XML document and use the UTF-8 encoding.

The following are examples of the XML document structure used for the SVC messages. The message type is identified by a combination of the <RequestCode> and <Reversal> elements. The first document is the SVC “reload void” message sent from POSI to iOS.

The <Amount> element in the void request message should include the amount sent in the original response message from iCare.

```
<?xml version=”1.0” encoding=”UTF-8”?>

<SVCMessage

  version=”1”

  posIntfc=”posInterfaceName”

posIntfcVersion=”1.00”
language=”eng-us”

  currency=”USD”

  sequence=”75”

  posSerialNum=”ABD12357875”

  retransmit=”n”>



  <RequestCode>RELOAD</RequestCode>

 <Reversal></Reversal>

  <TraceID>0312310000120211234</TraceID>

  <MID>ddddddddd</MID>

  <TerminalID>gggggggggg</TerminalID>

  <Amount>100.00</Amount>

 <LocalTime>015959</LocalTime>

 <LocalDate>20051231/</LocalDate>
```

```
  <CheckNumber>1234</CheckNumber>

 <BusinessDate>20051230</BusinessDate>

 <SVAN>7120567845749126</SVAN>

  <Track2>7120567845749126=1204</Track2>

 <TransactionEmployee>1234</TransactionEmployee>

</SVCMessage>
```

The next message is an example of a successful response to the reload void request.

```
<?xml version=”1.0” encoding=”UTF-8”?>

<SVCMessage

  version=”1”

language=”eng-us”
hostVersion=”1.00.8”

sequence=”75”>

  posSerialNum=”ABD12357875”

retransmit=”n”>



  <RequestCode>RELOAD</RequestCode>

 <Reversal></Reversal>

  <TraceID>03123100001202112 34</TraceID>

  <Amount>USDC000000002000</Amount>

 <BusinessDate>20051230</BusinessDate>

 <SVAN>7120567845749126</SVAN>

 <AuthorizationCode>ADVBCDE123</AuthorizationCode>

  <LocalBalance>USDC000000008000</LocalBalance>
  <AccountBalance>USDC000000008000</AccountBalance>

  <ExchangeRate>1.00</ExchangeRate>

  <ResponseCode>A</ResponseCode>

  <PrintLine> Check your account status at</PrintLine>

  <PrintLine>   www.myiCare.net</PrintLine>

</SVCMessage>
```

Successful Transmission

This section provides an example of a successful request/response transmission sequence between the POSI and iOS. Some required elements and attributes have been omitted for clarity.

ISSUE request message sent from POSI to iOS

```
<?xml version=”1.0” encoding=”UTF-8”?>

<SVCMessage

  version=”1”

  language=”eng-us”>

  currency=”USD”>

  <RequestCode>ISSUE</RequestCode>

  <TraceID>031114084910N042173</TraceID>

  …

</SVCMessage>
```

Response to ISSUE request from iOS to POSI.

```
<?xml version=”1.0” encoding=”UTF-8”?>

<SVCMessage

  version=”1”

  language=”eng-us”>
  currency=”USD”>

  <RequestCode>ISSUE</RequestCode>

  <TraceID>031114084910N042173</TraceID>

  <ResponseCode>A</ResponseCode>

  …

</SVCMessage>
```

Transmission Error Handling and Sequencing

This section discusses how error handling and transmission sequencing occur between the POSI and iOS are handled.

The iOS validates each received message. If a badly formed message is received, the iOS returns a Data Center error (“D” response code) along with an error message containing a description of the error. The following is an example response message:

```
<?xml version=”1.0” encoding=”UTF-8”?>

<SVCMessage hostVersion=”1.00.8” sequence=”04” retransmit=”N”
version=”1” language=”en-US” currency=”USD” >

<RequestCode>SV_ISSUE</RequestCode>
<TraceID>031114084910N042173</TraceID>

<Amount>23.00</Amount>

<ResponseCode hostCode=”20”>D</ResponseCode>

<DisplayMessage>Missing SVAN Element</DisplayMessage>

</SVCMessage>
```

Timeouts and Retransmissions

The POSI should establish a timeout for responses. It is suggested that 30 seconds be used for persistent IP connections and 60 seconds be used for dial-up implementations.

A retransmit attribute is included in each message. When the POSI initially sends a message it should include:

```
retransmit=”n”
```

as an attribute of the SVCMessage element.

If no response is received within the timeout period the POSI should retransmit the message and include:

retransmit=”y”

as an attribute of the SVCMessage element.

This indicates to the iOS that a retransmit has occurred and if it had already processed the initial message (i.e. the response was lost before reaching the POSI), it should just retransmit the response. This avoids duplicate message processing.

Message Sequencing

Each message includes a sequence attribute within the SvcMessage element. This is used to implement additional transaction fault tolerance. It also allows POS resets to be tracked.

The following process should be used:

1. The sequence number is unique to each POS terminal (not store).
2. On each POS terminal’s first message request, a sequence number of 0 should be used. This indicates to the iOS that the POS Terminal has been reset or powered off; resetting the iOS sequence number tracking for this POS Terminal.
3. The sequence number is incremented after each successful request/response cycle.
4. If a response is not received, the sequence number is NOT incremented.
5. When the sequence number is incremented past 99 it should be reset to 1.
6. If the POS is reset or power cycled the sequence number should be reset to 0.

When the iOS receives a request with a duplicate sequence number and the previous transaction was received less than 15 minutes ago, it assumes the previous response was not received and rolls back the previous transaction. If the previous transaction occurred more than 15 minutes previous, an exception condition will be logged at the data center.

3 SVC Message Overview

This section is divided into three parts:

1. SVC Transactional Message List: lists each of the available SVC messages, providing a brief description of the messages.
2. Attributes & Elements: describes each field type, attribute, and element type that may be present in a Request and Response message, including Check Summary.
3. Response Elements: describes response elements that may include POS instructions, such as Print Line Items, Action Items, and POS Items.

SVC Transaction Message List

Message	Request Message	Description
Issue SV Account	SV_ISSUE	This request is used to create an SV account with a balance.
Activate SV Account	SV_ACTIVATE	This request is used to activate an existing inactive SV account and associated preloaded balance.
Reload SV Account	SV_RELOAD	This request is used to credit an existing SV account.
SV Redemption Authorization	SV__REDEMPTION_AUTH	This request is used to authorize debiting of an existing SV account by a specified amount.
SV Redemption	SV__REDEMPTION	This request is used to debit an existing SV account by a specified amount.
SV Cash Out	SV_CASHOUT	This request is used to debit the remaining balance on an existing SV account.
SV Balance Inquiry	SV_BALANCE_INQUIRY	This request is used to obtain the account balance on an SV account.

Balance Transfer	SV_BALANCE_TRANSFER	This request is used to transfer the outstanding balance from on SV account to another SV account.
Issue Points	POINT_ISSUE	This request is used to add points to an existing account.
Point Balance Inquiry	POINT_BALANCE	This request is used to obtain the point balance for an account.
Accept Coupoon	SVC_ACCEPT_COUPON	This request is used to validate and accept a coupon.
Coupon Inquiry	COUPON_INQUIRY	This request is used to inquire about available coupons.
Issue Coupon	SV_ISSUE_COUPON	This request is used to issue a coupon.
Unique Item Inquiry	SV_UITEM_INQUIRY	This request is used to obtain the unique items purchased by an account. Requires Specific Loyalty configuration to track Unique Items.

Load Configuration Message List

Message	Request Message	Description
Load Configuration	LOAD_CFG	This request is used by the POS to provide specific configuration information.
Load Terminal Configuration	LOAD_TERM_CFG	This request is used by a standalone terminal, such as a VeriFone device, to return specific configuration.

See *Load Configuration* section for additional details.

Attributes and Elements

Field Type Definitions

Field Type	Description
A*x*	Refers to a text string where the "*x*" variable indicates the maximum number of characters in the field. For example, A16 means a field can contain up to 16 characters, while A1 means only one character is allowed in the field.
Date	Used for date data. Date is specified in YYYYMMDD format. Example: July 20, 2005 is specified as “20050720”
Decimal	Used for decimal or monetary data. Decimal data has the following format: [-][1-9][0-9]*[.][0-9]* The leading negative sign, ASCII value of 2D (‘-‘), is optional. Leading zeros are not permitted. The decimal is represented using the ASCII value of 2E (‘.’) and is optional if there is no fractional part for the value. Monetary Examples: 0-decimal currency: -99 2-decimal currency: -99.99 3-decimal currency: -99.999
Empty	Used for elements that don’t require any element data.
List	This field type is used for elements that contain a list of information. The list has a maximum character length of 4096, which makes it equivalent to an A4096 field. The individual values in the list are separated by the ‘|’ character (ASCII hexadecimal value 7C). For example, 123||abc|123 Specifies 4 elements, the second element is empty.
N*x*	Refers to a numeric string where the "*x*" variable indicates the maximum quantity of numbers in the field. For example, N1 means only one number is allowed in the field, while N2 means two numbers can be used in the field.

Time	Used for time data. Time is specified using the format HHMMSS. Example: 1:03:23 PM is specified as 130323. 12:00:00 AM is specified as 000000
XML	Used for elements that contain one or more XML elements as children.

Attribute Definitions

The following attributes may be present with each SVC Message element.

Attribute	Description	Type
version	This attribute defines the SVCMessage version. It is used to indicate to both iOS and POSI the format of the message and allows the potential interoperability of diverse iOS and POSI versions. It appears in both request and response messages.	A8
sequence	This attribute defines the message sequence number. On startup this number will begin at 0, and increment after each successful transaction. After 99 it will reset and start again at 1.	N2
retransmit	This attribute indicates if this message is a retransmission. It will contain either "n" or "y".	A1
language	This attribute indicates the native language used at the store such as "en-US" or "fr-FR".	A5
currency	This attribute indicates the native currency used by the store. The currency is represented using the 3 character alphabetic code specified in the ISO 4217 standard such as "USD" or "GBP".	A3
posIntfcName	This attribute is used in the request messages to indicate the name of the POSI.	A16
posIntfcVersion	This attribute is used in the request message to indicate the version of the POSI.	A8
hostVersion	This attribute is used in the response message to indicate the version of the iOS.	A8
posSerialNum	This optional attribute is used to provide the serial number of the POS terminal performing the request. It is typically used by CAT or VeriFone style terminals. When included it the request, it should be included in the response. Requires location to be configured as a Stand Alone location; Check Summary does not need to be present in request.	A16

The following attributes may be present with each ResponseCode element.

Attribute	Description	Type
hostCode	This indicates an error number and is used in conjunction with the error message. It may be used by support to more easily identify an error cause.	N4

The following attributes may be present with each LocalBalance element.

Attribute	Description	Type
programCode	The code / identifier for the program, matches the value of the "id" attribute of the programCode element.	A10
name	The name of the program (displayed / printed with the balance).	A32
type	The type of the program; (1) for currency, (2) for points.	N1
transAmount	The value applied in the current transaction. This attribute can be omitted in the inquire transaction.	Decimal

Element Definitions

The following elements are used in one or more request and/or response messages.

Element	Description	Type
AccountBalance	This element indicates the account balance in the data centers native currency.	Decimal
AccountCurrency	This element indicates the native currency used by the data center for this account. The currency is represented using the 3 character alphabetic code specified in the ISO 4217 standard. ISO 4217 currency codes can be looked up at http://en.wikipedia.org/wiki/ISO_4217 For example: USD is used for US dollar CAD is used for the Canadian dollar	A3
Actions	This element contains a list of actions that the user can perform in response to a request.	XML
AllTransCnt	This element indicates the number of SVC transaction included in the summary settlement message.	N9

AllTransTtl	This element indicates the total amount of all transactions in the summary settlement message. This value is represented in the local currency.	Decimal
Amount	This element indicates the value of the current transaction.	Decimal
AuthorizationCode	This element contains the authorization code created by the data center for the Redemption Authorization transaction.	A19
BonusPointsIssued	This value contains the number of bonus points issued.	Decimal
BusinessDate	This element indicates the business date of the transaction.	Date
CheckNumber	This element contains the check number or guest check identifier for the current POS check.	N10
CheckSummary	This element contains summary information about the current POS check.	XML
CreditTransCnt	This element indicates the number of credit transactions in the summary settlement.	N9
CreditTransTtl	This element indicates the total monetary value of all credit transactions in the summary settlement. This value is represented in the local currency.	Decimal
CouponCode	This element includes the coupon code.	A32
DebitTransCnt	This element indicates the number of debit transactions in the summary settlement.	N9
DebitTransTtl	This element indicates to total monetary value of all debit transactions in the summary settlement. This value is represented in the local currency.	Decimal
DayPart	Optional Day part, 1 – 9, indicating day part or serving period (breakfast, lunch, dinner, etc.).	N10
DisplayMessage	This element contains a message that is displayed to the user. The standard use of this element is to convey SVCHA/Data center errors to user when transactions have a ResponseCode of ‘E’, ‘V’, or ‘P’.	A100

ExchangeRate	This element indicates the exchange rate that is used to convert monetary amounts from the account to the local currency. If this element is omitted it has a default value of 1.00 (i.e., account and local currencies are the same). The currency conversion is performed depends on the currency configuration at the iOS ie:- (local currency) = (account currency) (* or /) (exchange rate)	Decimal
Extension	This element is used to add additional information to a message without requiring the message version to be updated. This element uses the “name” attribute to indicate the meaning of the element data. < Extension name=”address1”> ADDRESS EXAMPLE < /Extension> A message may contain zero or more Extension elements.	A100
ForeignCard	This element indicates if the SV card is a “Foreign” card. The element is only present if the card is a foreign card. It does not contain any data.	Empty
ItemNumber	This element is used the indication the number of the item applied to a check on a redemption transaction.	N9
LocalBalance	This element indicates the account balance in the local currency of the interface/POS system.	Decimal
LocalCurrency	This element indicates the native currency used by the data center for this account. Refer to the AccountCurrency element for additional details.	A3
LocalDate	This element indicates the local date of the transaction	Date
LocalTime	This element indicates the local time of the transaction.	Time
OldSVAN	This value contains an additional account number. It is used to perform operations on range of cards, or to transfer a balance between cards. When used to transfer balances, the old account or “from” account should be in this element and the new or “to” account should be in the SVAN element.	A19

OldTrack2	This value contains an additional track 2 data. It is used to perform operations on range of cards, or to transfer a balance between cards. The presence of this element indicates that the number was electronically entered. If the number is manually entered, then the OldSVAN element is used. See definition of Track2 element for additional details.	A40
PointsIssued	This value contains the number of points issued. This value will include bonus points issued also.	Decimal
PosItems (ItemType)	This element is used to indicate the type of POS item applied to a check after various transactions. The element may contain a list of POS items that may be applied to a POS check. The ItemType element can be one of the following values: ‘M’ – Menu Item or Sales Item ‘D’ – Discount ‘S’ – Service Charge ‘T’ – Tender Not all transactions will support all values. Refer to the transaction comments to determine which values are valid for each transaction. This element contains a list of POS items that can be applied to a POS check.	XML
PrintLine	This element contains one line of text that is printed by the POS client application. In addition to the text to be printed, formatting print-directives can be included. See The PrintLine Element for details on how the text is printed. The message may contain zero or more PrintLine elements.	A128
ProgramCode	A mnemonic describing the iCare program (e.g. GOLDCLUB, GIFT, PREMIER, HSE_ACCT).	A8
ProgramName	The customer-friendly name describing a program.	A32
RecordNum	This element is used to identify the transaction number of the SVC transaction being settled in the Transaction Settlement request.	N9
ReplaceSVAN	Flag used in balance transfer. If set, the SVAN is replaced in the existing accounts with the new SVAN rather than an original account closed and the balance added or transferred to the new account.	N1

RequestCode	This element is used to identify the type of transaction represented by the request document.	A100
ResponseCode	This element indicates the success or failure of the request. The following values are valid: ‘A’ = Approved ‘D’ = Data Center initiated error	A1
RevenueCenter	This element identifies the POS revenue center from which the request is originating.	N9
Reversal	This element is used to indicate that the transaction is a void/cancel of the normal RequestCode transaction. The element is omitted for normal RequestCode transaction. For a void/cancel transaction, the element is present, but contains no data.	Empty
SubstituteID	This element contains the substitute ID of account(s) for which the transaction is being executed. Not applicable to SV_ISSUE, SV_BALANCE_TRANSFER, SV_ACCEPT_COUPON, SV_ALT_ACCEPT_COUPON.	A32
SVAN	This element contains the account number for which the transaction is being executed.	A32
SVProgramCode	Optional element, is used to describe cards function when card may have multiple accounts and account is ambiguous based on the function. For example, Redemption function is ambiguous when using a card with both debit and award accounts. Message should include the SVProgramCode to resolve this.	A16
TerminalID	This element specifies the ID number of the POS terminal from which the request is originating.	N9
TerminalType	This element indicates the type of POS terminal from which the request is originating. Examples include: PCWS, WS4, SurePOS.	A8
TipAmount	This element specifies the portion of the Amount element that is allocated to a tip payment.	Decimal

TraceID	This element contains an auditing code that can be used to uniquely identify each transaction originating from a store system. It is possible for two stores to generate the same TraceID. This element has the format: YYMMDDHHMMSSRXXCCCC
	YY = current year (00-99
	MM = current month (01-12)
	DD = current day (01-31)
	HH = current hour (00-23)
	MM = current minute (00-59)
	SS = current second (00-59)
	R = Retransmit flag, "R" indicates a retransmit, while "N" indicates a new message
	XX = the message sequence number
	CCCC = Check number (1-9999)
		A25

Track1	This element contains the track 1 data when a SV card is read. This element is empty if no track 1 data exists.	A255
Track2	This element contains the track 2 data when a SV card is read. The format of the track data is defined in ISO 7811-1. Per this standard the track 2 data has a maximum length of 40 characters. The presence of this element indicates that the number was electronically entered. If the number is manually entered, then the SVAN element is used. This element does not contain the start and end sentinels of the card data, but does including the '=' field separator character between account and discretionary data fields. Because track data often XML markup characters, the track data is enclosed in <![CDATA[…]]> section.	A40
TransactionEmployee	This element identifies the employee performing the transaction.	N9

Check Summary Element

This section identifies the structure of the CheckSummary element. The CheckSummary element contains additional XML elements that provide information about sales itemizers, payments, and menu items on the current POS check. The following is an example of what the CheckSummary element might look like:

```
<CheckSummary>

  <SalesItemizers>
```

```
        <SI ID=”1”>6.00</SI>

        <SI ID=”2”>14.00</SI>

   </SalesItemizers>

   <MenuItems>

        <MI ID=”1” QTY="1">6.00</MI>

        <MI ID=”2” QTY="2">14.00</MI>

     </MenuItems>

   <Payments>

        <Payment ID=”101” QTY="1">1.00</Payment>

        <Payment ID=”222” QTY="1">1.00</Payment>

   </Payments>

   <Totals ttlTax="1.20"

         ttlDsc="0.00"

         ttlSvc="0.50"

         ttlPay="2.00"

         ttlDue="19.70">

</CheckSummary>
```

The <SalesItemizers> element is not a required; it contains a <SI> attribute for the number of sales itemizers supported by the POS system (if the POS supports 8 sales itemizers then the document contains 8 <SI> elements). Each ID attribute indicates the sales itemizer record number in the transaction. The element data indicates the amount of that sales itemizer record number.

The <MenuItems> element contains a <MI> attribute for each distinct menu item on the check. Each ID attribute is the menu item record number; the QTY attribute indicates the quantity of the MI ID in the transaction. The element data indicates the amount of the sales of that menu item record number in the transaction. This amount may be negative. The <MenuItems> element should be included in all CheckSummary elements, unless otherwise noted in this document or as instructed by MICROS.

The <Payments> element is not a required; it is only present if the check has 1 or more payments. Each ID attribute is the payment record number; the QTY attribute indicates the quantity of the Payment ID in the transaction. The element data indicates the payment amount associated payment record number in the transaction.

The <Totals> element includes various transactional summary totals:

ttlTax- The transaction tax total.
ttlDsc- The sum of all discounts associated with this transaction.
ttlSvc- The sum of all service charges associated with this transaction.
ttlPay- The sum of all payments associated with this transaction.
ttlDue- The total due from the customer or guest.

The element data for the <SI>, <MI>, <Payment>, <Totals> elements are expressed using the Decimal data type (see Field Type Definition for details).

Response Elements

PrintLine Element

The PrintLine element is used by the iOS to specify a line of text to be printed. The text may be printed on the POS workstation's printer as a chit or included as a transaction trailer.

The data being printed has no minimum length. Empty elements, or element with spaces should be printed as a blank line. Leading spaces should be preserved. The addition of leading spaces to a string may be used to center the text.

In addition to the text to be printed, several directives can be used to specify formatting instructions. A list of formatting directives is provided in the following table.

Name	Directive	Description
Doublewide On	&DWON;	Used to change the format of the text that follows to print as doublewide. Example: "&DWON;Hello world!" prints as "**H e l l o w o r l d !**"

Doublewide Off	&DWOFF;	Used to turn off doublewide formatting. Example: “&DWON;H&DWOFF;ello world!” prints as “**H** ello world!” A doublewide character for the letter “H” and normal characters for the remaining characters.
Red On	&RON;	Used to turn on the print-in-red formatting for the current line of text. Example: “Entire line is printed red.&RON;” prints as “Entire line is printed red.” **NOTE:** On a thermal printer, the &RON; directive results in text printing using white-on-black text. In addition, it is not possible to print only part of a line in red. The line “red off&RON; red on &ROFF; still prints the entire line using white-on-black text.
Red Off	&ROFF;	Used to turn off the print-in-red formatting for the current line of text. This directive does not have any effect when printing to a thermal printer.
Center	&C;	Centers the line of text by adding leading spaces. The center directive can appear any where in the line. Example: “`center me&C;`” prints as “ center me” Processing of the center directive results in 11 leading spaces being added to the line.

Actions Element

This section identifies the structure of the Actions element. The Actions element can appear in the response sent from the iCare Host. It defines one or more Actions the user can select from in the request response. For example, this element is used by the Coupon

Inquiry transaction to present the user with a list of coupons that can be accepted via the Accept Coupon transaction.

The following is an example of the Actions element in a response message:

```
<Actions>
  <Action>
    <Type tid="12" />
<Data pid="9">11230125</Data>
<Text>$5.00 Discount, Value: $5.00</Text>
</Action>
  <Action>
    <Type tid="12" />
<Data pid="9">11230208</Data>
<Text>Free Dessert Coupon, Value: $5.95</Text>
  </Action>
</Actions>
```

The <Actions> element is a container for one-or-more <Action> elements.

The <Action> element defines an action the user can choose to select in response to the request. Each action must contain a <Type>, <Data> and a <Text> element.

The <Type> element defines the type of transaction this action represents. The action is identified using the “tid” attribute. For example, the following action indicates an Accept Coupon transaction should be performed if the user selects this action:

```
    <Type tid="12" />
```

The <Data> element contains information that is passed back to the iOS if user selects this action. The pid attribute indicate the prompt/data that the subsequent transaction should return. See the pid table below for a list of supported prompts. For example, the following <Data> element is used to pass the value “11230125” in the <CouponCode> element. The value of the <Data> element is limited to 80-characters.

<Data pid="9">11230125</Data>

The <Text> element contains the text displayed to the user when they are asked to select from the list of actions. The value of the <Text> element is limited to 80-characters.

<Text>$5.00 Discount, Value: $5.00</Text>

Action Element: Transaction Identifiers (tid)

tid	Transaction Name
1	Issue
2	Activate
3	Reload
4	Cashout
5	Balance Inquire
6	Balance Transfer
7	Issue Points
8	Redeem Points
9	Points Inquire
10	Redeem
11	Redeem Authorization
12	Accept Coupon
13	Issue Batch
14	Activate Batch
15	Manual Redemption
17	Issue Coupon

Action Element: Prompt Identifiers (pid)

pid	Transaction Name
1	Please enter amount
2	Please swipe/enter card number
3	Please swipe/enter new card number
4	Please swipe/enter old card number
7	Please enter points amount
8	Please enter authorization code
9	Please enter coupon
10	Please select the program code
11	Please enter the PIN

PosItems Element

This section identifies the structure of the PosItems element. The PosItems element may appear in the response. It defines one or more POS items applied to a POS check. This element is used by the Issue Points transaction when issuing multiple discounts with multiple loyalty programs enabled per card rule.

The following is an example of the PosItems element in a response message:

```
<PosItems>

  <PosItem type="D" number="71">5.95</PosItem>

  <PosItem type="D" number="72">4.97</PosItem>

</PosItems>
```

The <PosItems> element is a container for one-or-more <PosItem> elements.

The <PosItem> element defines a POS item. The type attribute is the type of the item (see <ItemType> element). The number attribute is the number of the item (see <ItemNumber> element). The element value is the currency amount of the item to be applied to the check.

The order of the <PosItem> elements in the response is used to identify the order in which they are added to the check.

4 SVC Transactional Message Document

This section revisits each SVC Message, providing detailed information about the purpose and content of the transactional messages, covering the usage and composition of each SVC transactional request message, as well as its corresponding response message.

Presence of Element Key

For Request Documents and Response Documents, the options for the presence of an element are defined below.

Presence of Element	Description
Blank/Empty	Presence of this element is not relevant to the transaction.
Optional	May be included when necessary for a successful transaction.
Present	Should be included; primarily used for tracking purposes.
Required	Essential for a successful transaction.

It is suggested that Transaction Requests contain all elements indicated as Present or Required, as transactions may fail to complete.

Base Document – Transactional Request and Response Document

The content of all request and response documents are derived from the same base document. This base document is defined below.

Element	Request Document	Response Document
Request Code	Present <Transaction Specific>	Required <Transaction Specific>
Reversal	Optional	Required if present in the request
TraceID	Required	Required
TerminalID	Present	
TerminalType	Present – Not a predefined string	
Amount	Required	Required
LocalCurrency	Required	
LocalDate	Required	

LocalTime	Required	
BusinessDate	Required	Required
DayPart	Optional	
CheckNumber*	Optional	
TransactionEmployee	Required- If no Employee exists, may always be ‘1’ or an arbitrary number	
RevenueCenter	Required- If no Revenue Center exists, may always be ‘1’ or an arbitrary number	
SVProgramCode	Optional	
SVAN	Required- One of the following is required for all transactional messages: SVAN, Track1, Track2, or SubstituteID.	Required for all Transaction Messages
Track1		
Track2		
SubstituteID		
OldSVAN	Required- One of these is required for the Transfer Balance message: OldSVAN, OldTrack1, OldTrack2.	Required for Transfer Balance messages
OldTrack1		
OldTrack2		
AuthorizationCode		Required for Redemption Authorization
AccountCurrency		Required for all Transactional Messages
LocalBalance		Required for all Transactional Messages
AccountBalance		Required for all Transactional Messages
Exchange Rate		Required for all Transactional Messages
ResponseCode		Required
DisplayMessage		Optionally Returned
PrintLine		Optionally Returned
ProgramCode		Required for Transactional Messages
ProgramName		Required for Transactional Messages

* The CheckNumber element should be Present in transactions that require the CheckSummary to be Present, such as SV_Redemption and Point_Issue. See the SVC Transactional Messages section for additional details.

Response Overview – POS Item and Item Number

Various responses may return a POS Item type and Number. This may be used by the POSI to perform appropriate POS Transactions. For example, based on the iOS configuration, the response to an ISSUE request may be a Menu Item or Service Charge Item indicating to the POSI to add the SV Issue as a specific menu item or service charge to the current transaction.

The POSItem type element indicates the type of POS function to execute (menu or product item, discount, service charge, or tender). The item Number element indicates the specific POS record number to execute (menu or product item, discount, service charge or tender number).

The iOS configuration must be coordinated with the POS implementation or configuration so SVC debit, credit, and point issue transactions execute correctly.

Typically credit requests (issue, activate, reload, etc.) may include a menu item or service charge item in the response. Debit requests (redeem, cash out, etc.) may include discount or tender item in the response.

Typically point issue requests may include a $0 tender item or discount in the response. Accept Coupon requests may include a discount in the response.

Award accounts may be handled as a cash value account (similar to a gift card where a tender function may be used for redemption) or as a non-cash value account (similar to an award or promo card where a discount function is used for redemption).

This capability allows substantial program flexibility.

SVC Transactional Messages

Issue SV Account

Overview

This request may be used to issue one or more SV cards. When issuing multiple cards, the SVAN element is the first card number and the OldSVAN element is the last card number. The iOS will issue valid card numbers between SVAN and OldSVAN.

If a single SV Card is to be issued, the SV Account Number (SVAN) may be included with the request; alternatively it may also be automatically generated and returned by the iOS. In this case it is the POSI responsibility to print it for future reference.

Comments

The SVAN number should be retained for referencing for future transactions.

Valid ItemType values are ‘M’ and ‘S’.

Element	Request Document	Response Document
Request Code	SV_ISSUE	SV_ISSUE
ItemType		Required
ItemNumber		Required
<Base Document>		

Activate SV Account

Overview

This may be used to activate one or more SV cards. In this scenario, the card accounts will already be loaded at the Data Center and this process activates them.

This request may be used to activate one or more SV cards. When activating multiple cards, the SVAN element is the first card number and the OldSVAN element is the last card number.

Comments

Valid ItemType values are ‘M’ and ‘S’.

Element	Request Document	Response Document
Request Code	SV_ACTIVATE	SV_ACTIVATE
ItemType		Required
ItemNumber		Required
<Base Document>		

Reload SV Account

Overview

It may be used for recharging, reloading or adding additional value to SV Accounts.

Comments

Valid ItemType values are ‘M’ and ‘S’.

Element	Request Document	Response Document
Request Code	SV_RELOAD	SV_RELOAD
ItemType		Required
ItemNumber		Required
<Base Document>		

Redemption Authorization

Overview

This requests the authorization to debit an SV Account by an amount. The iOS may optional reserve an authorization amount. If the amount is reserved, it will be freed if not used with 8 hours. This request will return an authorization code.

This request may be issued when a guest tenders with the SV card and wishes to add a subsequent tip.

Comments

Valid ItemType value is 'T'.

Element	Request Document	Response Document
Request Code	SV_REDEMPTION_AUTH	SV_REDEMPTION_AUTH
CheckSummary	Present	Present
ItemType		Required
ItemNumber		Required
<Base Document>		

Redemption

Overview

This request may be used in two scenarios; the first is when a payment is to be applied without a prior authorization, the second is when a prior authorization has occurred. If a prior authorization occurred, this request frees the reserved amount; this request also debits the SV Account.

Comments

The response will contain an ItemType element, which may be applied to the transaction. Valid ItemType values are 'T' and 'D' (Tender and Discount).

In place of the account number, customer name can be used; enter any portion of lastname,firstname. First name is not required.
In place of the account number, phone number can be used; enter # followed by phone number. For example, #4435550100.

Element	Request Document	Response Document
Request Code	SV_REDEMPTION	SV_REDEMPTION
TipAmount	Present	
CheckSummary	Present	
ItemType		Required

ItemNumber		Required
<Base Document>		

Cash Out

Overview

This request is used when a customer wishes to cash out an existing SV Card. The iOS may be configured to disallow a cash out if card balance exceeds a configured value.

Comments

The response will contain an ItemType element, which may be applied to the transaction. Valid ItemType values are ‘T’ and ‘D’ (Tender and Discount).

Element	**Request Document**	**Response Document**
Request Code	SV_CASHOUT	SV_CASHOUT
ItemType		Required
ItemNumber		Required
<Base Document>		

Balance Inquiry

Overview

This request is used to obtain the account balance on an SV Account.

Comments

In place of the account number, customer name can be used; enter any portion of lastname,firstname. First name is not required.

In place of the account number, phone number can be used; enter # followed by phone number. For example, #4435550100.

Element	**Request Document**	**Response Document**
Request Code	SV_BALANCE_INQUIRY	SV_BALANCE_INQUIRY
<Base Document>		

Balance Transfer

Overview

This request is performed to transfer the account balance from one account number or card to another.

Comments

The OldSVAN element represents the "from" account and the SVAN element represents the "to" or new account.

Element	Request Document	Response Document
Request Code	SV_BALANCE_TRANSFER	SV_BALANCE_TRANSFER
<Base Document>		
<ReplaceSVAN>	Optional	

Issue Points

Overview

This request is performed when Loyalty Points should be issued to a Loyalty Card. The iOS may be configured with various rules that determine the point value added to an account or an award to be applied to the account, such as an immediate discount, coupon, or stored value. A CouponCode for redemption may be optionally included in the message.

Comments

In place of the account number, customer name can be used; enter any portion of lastname,firstname. First name is not required.
In place of the account number, phone number can be used; enter # followed by phone number. For example, #4435550100.

Element	Request Document	Response Document
Request Code	POINT_ISSUE	POINT_ISSUE
CheckSummary	Present	
CouponCode	Optional	
ItemType		Required (with single loyalty program per card)
ItemNumber		Required (with single loyalty program per card)
PosItems		Required (with single loyalty program per card)
<Base Document>		
BonusPointsIssued		Required
PointsIssued		Required

Prompt for SV Redemption or Prompt for Accept Coupon on Issue Points

Overview

This request may be performed when Loyalty Points should be issued to a Loyalty Card and the options 'prompt4RedeemOnIssuePoints' or 'prompt4CouponOnIssuePoints' are enabled. The point issue transaction is completed by three steps.

1) Inquire the available SV or the available coupons.

The iOS can use the value of the SVAN and CheckSummary to identify the available SV and the list of available coupons. Available SV or coupons are returned as Actions elements in the response.

2) Redeem SV or Accept Coupon
The POSI presents user a list of actions from which to select a choice. The POSI takes the information from the user's selection and performs the Redemption or the Accept Coupon transaction.

3) Issue Points

The final issue point request includes <IsFinal> element. iOS will issue points to the account.

Comments

Element	Request Document	Response Document
Request Code	POINT_ISSUE	POINT_ISSUE
CheckSummary	Present	
CouponCode	Optional	
ItemType		Required
ItemNumber		Required
<Base Document>		
BonusPointsIssued		Required
PointsIssued		Required
IsFinal	Present	
Actions		Present

SV Point Balance Inquiry

Overview

This request is used to retrieve a specific Loyalty Card's point balance.

Comments

In place of the account number, customer name can be used; enter any portion of lastname,firstname. First name is not required.

In place of the account number, phone number can be used; enter # followed by phone number. For example, #4435550100.

Element	Request Document	Response Document
Request Code	POINT_BALANCE	POINT_BALANCE
<Base Document>		

Accept Coupon

Overview

This request is used to accept a coupon. The coupon number should be placed in the CouponCode element. The coupon amount should be placed in the amount element. The coupon may be accepted or declined. A specific discount and tender item code will be returned.

Comments

Element	Request Document	Response Document
Request Code	SV_ACCEPT_COUPON	SV_ACCEPT_COUPON
CheckSummary	Present	
ItemType		Required ('D' or 'T')
ItemNumber		Required
<Base Document>		

Coupon Inquiry

Overview

This request is used to query the iOS for a list of coupons that can be applied. The iOS can use the value of the SVAN and CheckSummary to identify the list of available coupons.

Available coupons are returned as Actions elements in the response.
The POSI presents user a list of actions from which to select a choice. The POSI takes the information from the user's selection and performs the Accept Coupon transaction.

Comments

Element	Request Document	Response Document
Request Code	COUPON_INQUIRY	COUPON_INQUIRY
SVAN	Present	
CheckSummary	Present	
Actions		Present
<Base Document>		

Issue Coupon

Overview

This request is used to issue ad hoc coupons in POS. The issue coupon transaction is completed by two steps.

1) Inquiry the available coupons.

The request does not include CouponCode element, the iOS retrieve a list of POS ad hoc coupons from the database. The POS ad hoc coupons are the coupons that options 'allowPOSAdhocIssue' are enabled. The available coupons are returned as Actions elements in the response.

2) Issue a coupon

The POSI presents user a list of actions from which to select a choice. The POSI takes the information from the user's selection and performs the second Issue Coupon transudation (the request includes CouponCode element). The coupon is issued with the selected coupon code. A specific discount and tender item code will be returned.

Comments

Element	Request Document	Response Document
Request Code	SV_ISSUE_COUPON	SV_ISSUE_COUPON
SVAN	Present	
CouponCode	Present	
CheckSummary	Optional	
ItemType		Required ('D' or 'T')
ItemNumber		Required
Actions		Present
<Base Document>		

Unique Item Inquiry

Overview

This request is used to retrieve the Unique Items tracked for a specific Loyalty Card's point balance.

Comments

The inquiry is only available with specific iCare Loyalty configuration. For example, a Round Robin Program marketed as an 'Around the World' Beer Club—each unique purchases beer item is tracked per customer account.

Element	Request Document	Response Document
Request Code	SV_UITEM_INQUIRY	SV_UITEM_INQUIRY
<Base Document>		

Redeem Points

Overview

This request may be used when a payment is to be applied to the current transaction using points to pay a portion of the check. Points will be configured with a monetary balance in iCare, e.g. 1 point = $.25. This request debits the point balance of the Loyalty Card. Only one loyalty program on the Loyalty Account will support the redeem points function.

Comments

The response will contain an ItemType element, which may be applied to the transaction. Valid ItemType values are 'T' and 'D' (Tender and Discount).

Element	Request Document	Response Document
Request Code	POINT_REDEMPTION	POINT_REDEMPTION
TipAmount	Present	
CheckSummary	Present	
ItemType		Required
ItemNumber		Required
<Base Document>		

5 Load Configuration

Overview

This request is used by MICROS POS to request certain configuration information. The request should be initiated by the POS Solution.

Comments

Element	Request Document	Response Document
Request Code	LOAD_CFG	LOAD_CFG
<Base Document>		
<POS_Configuration>		Required

The response message Configuration element structure is:

Element | @Attribute	Field Type and Description	Example
POS_Configuration	Contains the configuration info	
@encryptedPIN	A1 - Optional: Indicates whether PINs are encrypted	“Y”
Options	One or more elements, one per RequestCode	
@rc	A16 - Required, contains Request Code	“POINT_ISSUE”
PromptForCoupon	A1 - Indicates whether user is to be prompted for Coupon with this request	“Y”
ProgramCode	A16- Optional, only present if user is to be promoted with Program	“Blue Club”
@id	A8 - Contains program code	“BCLUB”
@pinWithSwipe	A1 - Optional, set if pin is required when customer card is swiped	“Y”
@pinWithManEntry	A1 - Optional, set if pin is required when customer card number is manually entered	“Y”

6 Load Terminal Configuration

Overview

This request is used by a terminal to request terminal-specific configuration information. It is typically used by VeriFone style devices, i.e. Standalone Terminal Devices where there is no POS to send iCare transactions.

Comments

Standalone Terminal transactions should not include <MenuItems> in the CheckSummary Element. Item eligibility configured in iCare will be ignored when a transaction is received from a Standalone Terminal location.

Element	Request Document	Response Document
Request Code	LOAD_TERM_CFG	LOAD_TERM_CFG
<Base Document>		
<Terminal>	Required	
<Terminal_Configuration>		Required

The request message Terminal element structure is:

Element | @Attribute	Field Type and Description	Example
Terminal	Contains the terminal info	
@terminalVendor	A16 – Contains the terminal vendor's name	"VERIFONE"
@terminalModel	A16 – Contains the terminal model name	"Vx570"
@terminalVersion	A16 – Contains the terminal version	"V1.00"

The Terminal_Configuration element format is:

Element | @Attribute	Field Type and Description	Example
Terminal_Configuration	Contains the configuration info	
@terminalVendor	A16 – Contains the terminal vendor's name Same contents as in the request.	"VERIFONE"
@terminalModel	A16 – Contains the terminal model number Same contents as in the request.	"Vx570"

@terminalVersion	A16 – Contains the terminal version number Same contents as in the request.	“V1.00”
@localCurrency	A3 – Local currency represented using ISO 4217 Code (e.g. USD, GBP, EUR, CAD, etc.)	“USD”
@localLanguage	A5 -- Local lanuage to be used, represented using the W3C format for language codes (e.g. en-us, fr-ca, etc.)	“en-us”
@busDayStartTime	A5 – The business day start time, In 24 hour format (00:00 to 23:59). 4 am is 04:00, 1pm is 13:00.	“02:30”
@ipTimeout	N3 – The timeout interval in seconds for messages being transmitted over a persistent IP network.	“15”
@dialTimeout	N3 – The timeout in seconds for messages being transmitted over a dial network	“60”
@retryStrategy	N1 0 = No retry 1 = Send second message over same connection with retry flag set, 2 = Send second message over dial network with retry flag set.	“1”
@localConfigViewPwd	A16 – Password to be used for viewing configuration information at the terminal.	“Password”
@localConfigEditPwd	A16 – Password to be used when editing configuration information at the terminal.	“Password”
@reversalPwd	A16 – Password required when reversal transaction is performed.	“Password”
@encryptedPIN	A1 – Optional, if included and set to “Y”, indicates PINs are encrypted.	“Y”
LocationName	A16 – The store or location’s name.	“STORE”
TerminalName	A16 – The terminal name	“BAR”
ReceiptHeader	Contains the receipt header information. May include up to 5 Line elements.	
Line	A32 – Contains Line Description.	“WELCOME”
ReceiptTrailer	Contains the receipt trailer information. May include up to 5 Line elements.	

Line	A32 – Contains Line Description.	“Description of store”
Menu	Contains Menu Information	
MenuDescriptor	Contains information about one menu descriptor	
MenuPlacement	N1 – Menu Placement Contains the placement number (0-9)	“1”
Descriptor	A16 – Menu Descriptor	“GIFT”
LineFeedFunction-MenuPlacement	N2 – Linefeed function menu placement.	“1”
ReprintFunction-MenuPlacement	N2 – Reprint Receipt function menu placement.	“21”
DisplayVersionFunction-MenuPlacement	N2 – Display Version Function menu placement.	“20”
ReloadConfigFunction-MenuPlacement	N2 – Reload Configuration Menu placement	
Transactions	The element contains information about each transaction types configuration.	
Type	This element contains information about a specific transaction type. Note only active transaction types need to be included in the message.	

Name	A16 – This contains the name of the transaction type (It uses the request code except in the two lookup customer types **): “SV_ISSUE” – Issue Stored Value Account “SV_ACTIVATE” – Activate Stored Value Account “SV_RELOAD” – Reload Stored Value Account “SV_REDEMPTION_AUTH” – Authorize Redemption for Stored Value Account “SV_REDEMPTION” – Redeem Stored Value Account “SV_CASHOUT” – Cash-out Stored Value Account “SV_BALANCE_INQUIRY” – Inquire balance from Stored Value Account “SV_BALANCE_TRANSFER” – Transfer balance from one account to another. “POINT_ISSUE” – Issue Loyalty Points- If # included before number, will assume phone number, not account number, e.g.# 4435550100 “POINT_BALANCE” – Inquire Point Balance from Account. “CUSTOMER_LOOKUP_BY_PHONE” – Lookup Customer by phone number ** “CUSTOMER_LOOKUP_BY_NAME” – Lookup Customer by Name ** “SV_ACCEPT_COUPON” – Accept Coupon	
MenuDescriptor	A16 – This is this Transaction Types menu descriptor	“ISSUE CARD”
MenuPlacement	N2 – This is this transaction type’s menu placement.	“22”
@enable	A1 – Indicates whether this transaction type is enabled. (“N” or “Y”).	“Y”
@promptForRevCtr	A1 – Indicates whether the terminal will prompt for a revenue center during this transaction type. (“N” or “Y”)	“Y”

@promptForCouponCode	A1 – Indicates whether the terminal will prompt for a Coupon Code during this transaction type. (“N” or “Y”)	“Y”
@requirePwdForReversal	A1 – Indicates whether the terminal will prompt for a revenue center during this transaction type. (“N” or “Y”)	“Y”
@requirePINWithSwipe	A1 – Indicates whether this terminal will prompt for PIN entry when the Member Card is swiped. (“N” or “Y”)	“N”
@requirePINWithManEntry	A1 – Indicates whether this terminal will prompt for PIN entry when the Member Account Number is manually entered. (“N” or “Y”).	“N”
Programs	This element contains information about each program.	
Program	This element contains information about a specific program.	
@type	N1 – Defines program type: 1 = Debit Card or Gift Card Program 2 = Loyalty or points based program 3 = Credit Card Program 4 = Award Programs	“1”
@name	A16 – Program name	“Gift Card”
@code	A16 – Program Code	“GIFT”
RevenueCenters	This element contains information about each Revenue Center.	
RevenueCenter	This element contains information about a specific program.	
@name	A16 – Contains the Revenue Center’s Name	“CURB-SIDE”
@number	N9 – Contains the Revenue Center’s Number.	“122345”

7 Appendix I: Web Service Definition Language (WSDL) Document

StoredValueService.xml

```
<definitions name='StoredValueService'
targetNamespace='ejb.storedValue.micros.com'
xmlns='http://schemas.xmlsoap.org/wsdl/'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:tns='ejb.storedValue.micros.com'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'>

<types></types>

<message name='StoredValueWSEndPoint_processRequest'>

<part name='in0' type='xsd:string'></part>

<part name='in1' type='xsd:string'></part>

<part name='in2' type='xsd:string'></part>

<part name='in3' type='xsd:string'></part>

</message>

<message name='StoredValueWSEndPoint_processERequest'>

<part name='in0' type='xsd:string'></part>

<part name='in1' type='xsd:string'></part>

<part name='in2' type='xsd:string'></part>

<part name='in3' type='xsd:string'></part>

</message>

<message name='StoredValueWSEndPoint_processRequestResponse'>

<part name='result' type='xsd:string'></part>
```

```
</message>

<message name='StoredValueWSEndPoint_processERequestResponse'>

<part name='result' type='xsd:string'></part>

</message>

<message name='StoredValueWSEndPoint_processCRequestResponse'>

<part name='result' type='xsd:string'></part>

</message>

<message name='StoredValueWSEndPoint_processCRequest'>

<part name='in0' type='xsd:string'></part>

<part name='in1' type='xsd:string'></part>

<part name='in2' type='xsd:string'></part>

<part name='in3' type='xsd:string'></part>

</message>

<portType name='StoredValueWSEndPoint'>

<operation name='processCRequest' parameterOrder='in0 in1 in2 in3'>

<input message='tns:StoredValueWSEndPoint_processCRequest'></input>

<output
message='tns:StoredValueWSEndPoint_processCRequestResponse'></output>

</operation>

<operation name='processERequest' parameterOrder='in0 in1 in2 in3'>

<input message='tns:StoredValueWSEndPoint_processERequest'></input>

<output
message='tns:StoredValueWSEndPoint_processERequestResponse'></output>

</operation>
```

```
<operation name='processRequest' parameterOrder='in0 in1 in2 in3'>

<input message='tns:StoredValueWSEndPoint_processRequest'></input>

<output
message='tns:StoredValueWSEndPoint_processRequestResponse'></output>

</operation>

</portType>

<binding name='StoredValueWSEndPointBinding'
type='tns:StoredValueWSEndPoint'>

<soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http'/>

<operation name='processCRequest'>

<soap:operation soapAction="/>

<input>

<soap:body namespace='ejb.storedValue.micros.com' use='literal'/>

</input>

<output>

<soap:body namespace='ejb.storedValue.micros.com' use='literal'/>

</output>

</operation>

<operation name='processERequest'>

<soap:operation soapAction="/>

<input>

<soap:body namespace='ejb.storedValue.micros.com' use='literal'/>

</input>

<output>
```

```
<soap:body namespace='ejb.storedValue.micros.com' use='literal'/>

</output>

</operation>

<operation name='processRequest'>

<soap:operation soapAction="/>

<input>

<soap:body namespace='ejb.storedValue.micros.com' use='literal'/>

</input>

<output>

<soap:body namespace='ejb.storedValue.micros.com' use='literal'/>

</output>

</operation>

</binding>

<service name='StoredValueService'>

<port binding='tns:StoredValueWSEndPointBinding'
name='StoredValueWSEndPointPort'>

<soap:address
location='https://undefined.host:9443/ws/services/StoredValueService'/>

</port>

</service>

</definitions>
```

8 Appendix II: Example SOAP Trace

The following SOAP trace illustrates a typical request and response.

Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:ns="ejb.storedValue.micros.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Body SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns:processRequest id="o0" SOAP-ENC:root="1">
   <in0 xsi:type="xsd:string">&lt;?xml version="1.0"?&gt;
&lt;SVCMessage version="1" timeout="20" language="en_US" currency="USD"
sequence="5"&gt;&lt;RequestCode&gt;SV_ISSUE&lt;/RequestCode&gt;&lt;Tra
ceID&gt;030819004002000000024321&lt;/TraceID&gt;&lt;TerminalID&gt;99
&lt;/TerminalID&gt;&lt;TerminalType&gt;OTS&lt;/TerminalType&gt;&lt;Amo
unt&gt;138.23&lt;/Amount&gt;&lt;CheckNumber&gt;0073&lt;/CheckNumber&
gt;&lt;AuthorizationCode&gt;6611114&lt;/AuthorizationCode&gt;&lt;LocalCurr
ency&gt;USD&lt;/LocalCurrency&gt;&lt;LocalDate&gt;20030819&lt;/LocalDat
e&gt;&lt;LocalTime&gt;004002&lt;/LocalTime&gt;&lt;BusinessDate&gt;20030
819&lt;/BusinessDate&gt;&lt;TransactionEmployee&gt;123&lt;/TransactionEmp
loyee&gt;&lt;RevenueCenter&gt;2&lt;/RevenueCenter&gt;&lt;SVAN&gt;44443
33322221111&lt;/SVAN&gt;&lt;Track1&gt;&lt;/Track1&gt;&lt;Track2&gt;&lt;/
Track2&gt;&lt;OldS
VAN&gt;&lt;/OldSVAN&gt;&lt;OldTrack1&gt;&lt;/OldTrack1&gt;&lt;OldTrac
k2&gt;&lt;/OldTrack2&gt;&lt;/SVCMessage&gt;</in0>
   <in1 xsi:type="xsd:string">store1</in1>
   <in2 xsi:type="xsd:string">password</in2>
   <in3 xsi:type="xsd:string">98A83356</in3>
  </ns:processRequest>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <ns1:processRequestResponse
```

```
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="ejb.storedValue.micros.com">
   <ns1:processRequestReturn xsi:type="xsd:string">&lt;?xml
version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;&#xd;
&lt;SVCMessage version=&quot;1&quot; timeout=&quot;20&quot;
language=&quot;en_US&quot; currency=&quot;USD&quot;
sequence=&quot;5&quot;&gt;&lt;RequestCode&gt;SV_ISSUE&lt;/RequestCode
&gt;&lt;TraceID&gt;030819004002000000024321&lt;/TraceID&gt;&lt;Respon
seCode&gt;N&lt;/ResponseCode&gt;&lt;/SVCMessage&gt;&#xd;
</ns1:processRequestReturn>
  </ns1:processRequestResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

9 Appendix III: Manual Authorization Validation

In situations where communications cannot be established with the IOS a manual authorization may be preformed.

If a manual Redemption authorization is performed (by calling a customer service center or toll free IVR solution), the authorization number may be validated at the POS by the POSI.

If this is required the POSI should implement the validation algorithm. This algorithm uses the account number, transaction amount and check number to validate the authorization code. The algorithm specific may be requested from MICROS.

10 Appendix IV: CRC value calculation

The following three steps should be followed to derive the CRC value for <in3> of the defined SOAP request.

1. Take the original request message before xml escape and remove "\r" if any.
2. Take the CRC32 value of the message string from 1.
3. Convert the integer from 2 to an upper case string of ASCII digits in hexadecimal (base 16) with no extra leading **0**s.

For example, the following is the original request message in Appendix III.

```
<?xml version="1.0"?>

<SVCMessage version="1" timeout="20" language="en_US" currency="USD"
sequence="5"><RequestCode>SV_ISSUE</RequestCode><TraceID>030819004
0020000000024321</TraceID><TerminalID>99</TerminalID><TerminalType>
OTS</TerminalType><Amount>138.23</Amount><CheckNumber>0073</Chec
kNumber><AuthorizationCode>6611114</AuthorizationCode><LocalCurrency>
USD</LocalCurrency><LocalDate>20030819</LocalDate><LocalTime>004002
</LocalTime><BusinessDate>20030819</BusinessDate><TransactionEmployee
>123</TransactionEmployee><RevenueCenter>2</RevenueCenter><SVAN>444
4333322221111</SVAN><Track1></Track1><Track2></Track2><OldSVAN><
/OldSVAN><OldTrack1></OldTrack1><OldTrack2></OldTrack2></SVCMessa
ge>
```

Note that the message contains only one carriage return right after the first line.

Going through the above three steps, the result string should be 98A83356, which should be put in <in3>.

11 Java Sample Code

```
import java.util.zip.CRC32;

…

        String requestXml = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"+

                              "<CRMMessage language=\"en_US\"
currency=\"USD\">\n"+

                              "
<RequestCode>GetColumnList</RequestCode>\n"+

                              " <QueryCriteria request=\"Customer\" />\n"+

                              "</CRMMessage>";

        System.out.p
```