Sage Exchange Desktop v2.0 XML Messaging

Sage Payment Solutions

May 2017

© 2016 The Sage Group plc or its licensors. All rights reserved. Sage, Sage logos, and Sage product and service names mentioned herein are the trademarks of The Sage Group plc or its licensors. All other trademarks are the property of their respective owners.

Sage End User License Agreement (EULA): <http://na.sage.com/sage-na/eula>

Contents

[Revisions 6](#_Toc482590051)

[XML Schema 7](#_Toc482590052)

[Request 7](#_Toc482590053)

[Response 14](#_Toc482590054)

[XML Reserved Characters 19](#_Toc482590055)

[Methods 19](#_Toc482590056)

[Message formats 21](#_Toc482590057)

[Sale 21](#_Toc482590058)

[XML sale request with user interface sample 21](#_Toc482590059)

[XML sale request without user interface sample 22](#_Toc482590060)

[XML sale request and vault storage with user interface sample 23](#_Toc482590061)

[Authorization 24](#_Toc482590062)

[XML authorization request with user interface sample 24](#_Toc482590063)

[XML authorization request without user interface sample 25](#_Toc482590064)

[Capture 26](#_Toc482590065)

[XML capture request with user interface sample 26](#_Toc482590066)

[XML capture request without user interface sample 27](#_Toc482590067)

[Force 28](#_Toc482590068)

[XML force request with user interface sample 28](#_Toc482590069)

[XML force request without user interface sample 29](#_Toc482590070)

[Level 2 30](#_Toc482590071)

[XML Level 2 sale request with user interface sample 30](#_Toc482590072)

[Void 31](#_Toc482590073)

[XML void request without user interface sample 31](#_Toc482590074)

[Credit 32](#_Toc482590075)

[XML credit request with user interface sample 32](#_Toc482590076)

[XML credit request without user interface sample 32](#_Toc482590077)

[Single payment response 33](#_Toc482590078)

[XML payment response 33](#_Toc482590079)

[Batch inquiry 35](#_Toc482590080)

[XML batch inquiry request 35](#_Toc482590081)

[Batch close 35](#_Toc482590082)

[XML batch close request with net and count verification 36](#_Toc482590083)

[XML batch close request without net and count verification 36](#_Toc482590084)

[Vault operation 36](#_Toc482590085)

[XML vault operation request for creating a storage record 36](#_Toc482590086)

[XML vault operation request for updating a storage record 37](#_Toc482590087)

[Transaction status query 37](#_Toc482590088)

[XML transaction status query 38](#_Toc482590089)

[Vault status query 38](#_Toc482590090)

[XML vault status query 38](#_Toc482590091)

[Multi-payment processing 39](#_Toc482590092)

[XML multi-payment request with user interface 39](#_Toc482590093)

[Split payment processing 40](#_Toc482590094)

[XML split payment request with user interface 40](#_Toc482590095)

[Terminal item list 43](#_Toc482590096)

[XML terminal items list request 43](#_Toc482590097)

[Healthcare signature 43](#_Toc482590098)

[XML healthcare signature request 43](#_Toc482590099)

[XML healthcare signature response 44](#_Toc482590100)

[Tender type 45](#_Toc482590101)

[XML credit card sale request with user interface sample 45](#_Toc482590102)

[XML cash sale request with user interface sample 46](#_Toc482590103)

[EMV Receipt Request 47](#_Toc482590104)

[XML EMV Receipt Request Sample 47](#_Toc482590105)

[Terminal debit configuration 47](#_Toc482590106)

[XML Credit Card Sale Request with User Interface and Terminal Debit Configuration Sample 47](#_Toc482590107)

[Healthcare 48](#_Toc482590108)

[XML healthcare sale request with user interface sample 49](#_Toc482590109)

[Request field definitions 51](#_Toc482590110)

[Request\_V1 Type element - required 51](#_Toc482590111)

[ApplicationType element - required 51](#_Toc482590112)

[Payments element - optional 52](#_Toc482590113)

[Batch element - optional 52](#_Toc482590114)

[BatchType element 52](#_Toc482590115)

[HealthcareSignatureType element - optional 53](#_Toc482590116)

[PaymentType element 54](#_Toc482590117)

[MerchantType element 55](#_Toc482590118)

[TransactionBaseType element 56](#_Toc482590119)

[Customer element: PersonType element 57](#_Toc482590120)

[ShippingRecipient element: PersonType element 58](#_Toc482590121)

[Level2Type element 58](#_Toc482590122)

[Level3Type element 59](#_Toc482590123)

[Level3LineItemType element 60](#_Toc482590124)

[VaultStorageType element 61](#_Toc482590125)

[PostbackType element 63](#_Toc482590126)

[Response field definitions 65](#_Toc482590127)

[PaymentResponseType element - required 65](#_Toc482590128)

[ResponseType element - required 66](#_Toc482590129)

[VaultResponseType element 66](#_Toc482590130)

[RecurringResponseType element 67](#_Toc482590131)

[TransactionResponseType element 68](#_Toc482590132)

[EmvResponseType element 70](#_Toc482590133)

[MonerisReceiptType element 71](#_Toc482590134)

[BatchResponseType element 74](#_Toc482590135)

[TransactionSettlementStatusType element 75](#_Toc482590136)

[TransactionStatusQueryResponseType element 75](#_Toc482590137)

[RecurringStatusQueryResponseType element 76](#_Toc482590138)

[VaultAccountResponseType element 77](#_Toc482590139)

[AccountQueryResponseType Element 78](#_Toc482590140)

[Field validation 79](#_Toc482590141)

[Alpha numeric fields\* 79](#_Toc482590142)

[Numeric fields\* 79](#_Toc482590143)

[Gateway error codes 80](#_Toc482590144)

Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Date | Revision | Author |
| 1.00a | 10/1/2013 | Created document branch | Martin Henderson |
| 1.001 | 10/18/2013 | Added Tender Type section | Martin Henderson |
| 1.02 | 10/25/2013 | Healthcare Signature element renamed, Added Terminal Debit Configuration section | Martin Henderson |
| 1.03 | 10/31/2013 | Added PrerscriptionAmount element in Healthcare XML request. Added Payment response XML section, corrected Healthcare Signature element typo. Added Healthcare Signature response XML section. | Martin Henderson |
| 1.04 | 10/31/2013 | Added Postback request element section | Martin Henderson |
| 1.05 | 2/11/2014 | Corrected typo in Terminal List Item XML | Martin Henderson |
| 1.06 | 4/20/2016 | Updated with new EMV fields. Re-added missing response section. Re-did schema. | Majid Razvi |
| 1.07 | 4/25/2016 | Minor changes to account for response class refactoring. | Majid Razvi |
| 1.08 | 10/21/2016 | Added “EMV Receipts” category and an XML sample for “Request of an Existing EMV Receipt”.  Updated XML schema:   * Added EMVReceiptQuery element and EMVReceiptQueryType to Request\_v1 * Added EMVReceiptQueryResponse element, EMVReceiptQueryResponseType and EMVReceiptData type to Response\_v1 * Added EmvReceipt element to PaymentResponseType | Alex Nagoga |
| 1.09 | 5/15/2017 | Updated with new Moneris receipt type. | Alex Nagoga |
| 1.10 | 6/16/2017 | Added CanEnterAccount element | Alex Nagoga |

XML Schema

Request

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="Request\_v1" nillable="true" type="Request\_v1" />  <xs:complexType name="Request\_v1">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Application" type="ApplicationType" />  <xs:element minOccurs="0" maxOccurs="1" name="Payments" type="ArrayOfPaymentType" />  <xs:element minOccurs="0" maxOccurs="1" name="Batch" type="BatchType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionStatusQueries" type="ArrayOfTransactionStatusQueryType" />  <xs:element minOccurs="0" maxOccurs="1" name="RecurringStatusQueries" type="ArrayOfRecurringStatusQueryType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultStatusQuery" type="VaultStatusQueryType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultOperation" type="VaultOperationType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultAccount" type="VaultAccountType" />  <xs:element minOccurs="0" maxOccurs="1" name="UI" type="UIType" />  <xs:element minOccurs="0" maxOccurs="1" name="AccountQuery" type="AccountQueryType" />  <xs:element minOccurs="0" maxOccurs="1" name="Postback" type="PostbackType" />  <xs:element minOccurs="1" maxOccurs="1" name="IsSplitPayment" type="xs:boolean" />  <xs:element minOccurs="0" maxOccurs="1" name="HealthcareSignature" type="HeathcareSignatureType" />  <xs:element minOccurs="0" maxOccurs="1" name="TerminalItemListDisplay" type="TerminalItemListDisplayType" />  <xs:element minOccurs="0" maxOccurs="1" name="TerminalDebitConfiguration" type="TerminalDebitConfigurationType" />  <xs:element minOccurs="0" maxOccurs="1" name="EMVReceiptQuery" type="EMVReceiptQueryType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ApplicationType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="ApplicationID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="LanguageID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ClientID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ClientKey" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfPaymentType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="PaymentType" nillable="true" type="PaymentType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="PaymentType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Client" type="ClientType" />  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionBase" type="TransactionBaseType" />  <xs:element minOccurs="0" maxOccurs="1" name="Customer" type="PersonType" />  <xs:element minOccurs="0" maxOccurs="1" name="ShippingRecipient" type="PersonType" />  <xs:element minOccurs="0" maxOccurs="1" name="Level2" type="Level2Type" />  <xs:element minOccurs="0" maxOccurs="1" name="Level3" type="Level3Type" />  <xs:element minOccurs="0" maxOccurs="1" name="Recurring" type="RecurringType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultStorage" type="VaultStorageType" />  <xs:element minOccurs="0" maxOccurs="1" name="Postback" type="PostbackType" />  <xs:element minOccurs="0" maxOccurs="1" name="Healthcare" type="HealthcareType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Key" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ID" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="MerchantType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="MerchantID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantKey" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TransactionBaseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="TransactionID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionType" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Reference1" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Reference2" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Amount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="AuthCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="VANReference" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="CustomerType" type="CustomerType" />  <xs:element minOccurs="1" maxOccurs="1" name="TenderType" type="TransactionTenderType" />  </xs:sequence>  </xs:complexType>  <xs:simpleType name="CustomerType">  <xs:restriction base="xs:string">  <xs:enumeration value="Business" />  <xs:enumeration value="Consumer" />  <xs:enumeration value="Government" />  <xs:enumeration value="Undetermined" />  </xs:restriction>  </xs:simpleType>  <xs:simpleType name="TransactionTenderType">  <xs:restriction base="xs:string">  <xs:enumeration value="CREDITCARD" />  <xs:enumeration value="VIRTUALCHECK" />  <xs:enumeration value="CASH" />  <xs:enumeration value="UNDETERMINED" />  </xs:restriction>  </xs:simpleType>  <xs:complexType name="PersonType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="NameType" />  <xs:element minOccurs="0" maxOccurs="1" name="Address" type="AddressType" />  <xs:element minOccurs="0" maxOccurs="1" name="Company" type="CompanyType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="NameType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="FirstName" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MI" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="LastName" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="AddressType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="AddressLine1" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AddressLine2" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="City" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="State" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ZipCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Country" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="EmailAddress" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Telephone" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Fax" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="CompanyType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Address" type="AddressType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="Level2Type">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="CustomerNumber" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TaxAmount" type="xs:double" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="Level3Type">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Level2" type="Level2Type" />  <xs:element minOccurs="1" maxOccurs="1" name="ShippingAmount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="DestinationZipCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="DestinationCountryCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="VATNumber" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="DiscountAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="DutyAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="NationalTaxAmount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="VATInvoiceNumber" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="VATTaxAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="VATTaxRate" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="LineItems" type="ArrayOfLevel3LineItemType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfLevel3LineItemType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="Level3LineItemType" nillable="true" type="Level3LineItemType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="Level3LineItemType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="CommodityCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ProductCode" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Quantity" type="xs:int" />  <xs:element minOccurs="0" maxOccurs="1" name="UnitOfMeasure" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="UnitCost" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="TaxAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="TaxRate" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="DiscountAmount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="AlternateTaxIdentifier" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TaxTypeApplied" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="DiscountIndicator" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="NetGrossIndicator" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="ExtendedItemAmount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="DebitCreditIndicator" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="RecurringType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Schedule" type="ScheduleType" />  <xs:element minOccurs="1" maxOccurs="1" name="Interval" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="DayOfMonth" type="xs:int" />  <xs:element minOccurs="0" maxOccurs="1" name="StartDate" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Amount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="TimesToProcess" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="NonBusinessDay" type="NonBusinessDayType" />  </xs:sequence>  </xs:complexType>  <xs:simpleType name="ScheduleType">  <xs:restriction base="xs:string">  <xs:enumeration value="MONTHLY" />  <xs:enumeration value="DAILY" />  </xs:restriction>  </xs:simpleType>  <xs:simpleType name="NonBusinessDayType">  <xs:restriction base="xs:string">  <xs:enumeration value="THATDAY" />  <xs:enumeration value="BEFORE" />  <xs:enumeration value="AFTER" />  </xs:restriction>  </xs:simpleType>  <xs:complexType name="VaultStorageType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="GUID" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Service" type="VaultServiceType" />  </xs:sequence>  </xs:complexType>  <xs:simpleType name="VaultServiceType">  <xs:restriction base="xs:string">  <xs:enumeration value="CREATE" />  <xs:enumeration value="UPDATE" />  <xs:enumeration value="RETRIEVE" />  <xs:enumeration value="DELETE" />  </xs:restriction>  </xs:simpleType>  <xs:complexType name="PostbackType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="HttpsUrl" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="HealthcareType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="HealthcareAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="ClinicAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="DentalAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="VisionAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="PerscriptionAmount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="IIASVerification" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="HealthcareItems" type="ArrayOfHealthcareItemType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfHealthcareItemType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="HealthcareItemType" nillable="true" type="HealthcareItemType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="HealthcareItemType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Quantity" type="xs:int" />  <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Price" type="xs:double" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="BatchType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="1" maxOccurs="1" name="Net" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="Count" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="BatchPayment" type="BatchPaymentType" />  </xs:sequence>  </xs:complexType>  <xs:simpleType name="BatchPaymentType">  <xs:restriction base="xs:string">  <xs:enumeration value="CREDITCARD" />  <xs:enumeration value="PURCHASECARD" />  <xs:enumeration value="VIRTUALCHECK" />  </xs:restriction>  </xs:simpleType>  <xs:complexType name="ArrayOfTransactionStatusQueryType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="TransactionStatusQueryType" nillable="true" type="TransactionStatusQueryType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TransactionStatusQueryType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionID" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfRecurringStatusQueryType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="RecurringStatusQueryType" nillable="true" type="RecurringStatusQueryType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="RecurringStatusQueryType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="RecurringID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="StartDate" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="EndDate" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultStatusQueryType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultID" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultOperationType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="VaultID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultStorage" type="VaultStorageType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultAccountType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Company" type="CompanyType" />  <xs:element minOccurs="0" maxOccurs="1" name="Contact" type="PersonType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="SinglePayment" type="SinglePaymentUIType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultOperation" type="VaultOperationUIType" />  <xs:element minOccurs="0" maxOccurs="1" name="UIStyle" type="UIStyleType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="SinglePaymentUIType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="TransactionBase" type="UITransactionBaseType" />  <xs:element minOccurs="0" maxOccurs="1" name="Customer" type="UIPersonType" />  <xs:element minOccurs="1" maxOccurs="1" name="PostAuthorizationAnalysis" type="xs:boolean" />  <xs:element minOccurs="1" maxOccurs="1" name="CVVPrompt" type="xs:boolean" />  <xs:element minOccurs="1" maxOccurs="1" name="CanEditAccount" type="xs:boolean" />  <xs:element minOccurs="1" maxOccurs="1" name="CanUseExternalDevice" type="xs:boolean" />  <xs:element minOccurs="1" maxOccurs="1" name="CanEnterAccount" type="xs:boolean" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UITransactionBaseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Reference1" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="Amount" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="TaxAmount" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="ShippingAmount" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="AuthCode" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="CustomerNumber" type="UIFieldType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIFieldType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Enabled" type="xs:boolean" />  <xs:element minOccurs="1" maxOccurs="1" name="Visible" type="xs:boolean" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIPersonType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="UINameType" />  <xs:element minOccurs="0" maxOccurs="1" name="Address" type="UIAddressType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UINameType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="FirstName" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="MI" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="LastName" type="UIFieldType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIAddressType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="AddressLine1" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="AddressLine2" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="City" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="State" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="ZipCode" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="Country" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="EmailAddress" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="Telephone" type="UIFieldType" />  <xs:element minOccurs="0" maxOccurs="1" name="Fax" type="UIFieldType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultOperationUIType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="AccountNumber" type="UIFieldType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIStyleType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Wizard" type="UIElementStyleType" />  <xs:element minOccurs="0" maxOccurs="1" name="WizardTitle" type="UIElementStyleType" />  <xs:element minOccurs="0" maxOccurs="1" name="WizardStepLeft" type="UIElementStyleType" />  <xs:element minOccurs="0" maxOccurs="1" name="WizardStepRight" type="UIElementStyleType" />  <xs:element minOccurs="0" maxOccurs="1" name="WizardSupport" type="UIWizardSupportStyle" />  <xs:element minOccurs="0" maxOccurs="1" name="Buttons" type="UIElementStyleType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIElementStyleType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="BorderStyle" type="UIBorderStyleType" />  <xs:element minOccurs="0" maxOccurs="1" name="BackgroundColor" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="LabelStyle" type="UIFontStyleType" />  <xs:element minOccurs="0" maxOccurs="1" name="FieldStyle" type="UIFontStyleType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIBorderStyleType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="BorderTop" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="BorderBottom" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="BorderLeft" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="BorderRight" type="xs:int" />  <xs:element minOccurs="0" maxOccurs="1" name="BorderColor" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIFontStyleType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Family" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Color" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Size" type="xs:int" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="UIWizardSupportStyle">  <xs:complexContent mixed="false">  <xs:extension base="UIElementStyleType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Visible" type="xs:boolean" />  </xs:sequence>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="AccountQueryType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="HeathcareSignatureType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="VANReference" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SignatureAttributes" type="ArrayOfSignatureAttributeType" />  <xs:element minOccurs="0" maxOccurs="1" name="ItemsDisplayText" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Items" type="ArrayOfString" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfSignatureAttributeType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="SignatureAttributeType" nillable="true" type="SignatureAttributeType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="SignatureAttributeType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="DisplayText" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfString">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TerminalItemListDisplayType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="TerminalDisplayItems" type="ArrayOfTerminalDisplayItemType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfTerminalDisplayItemType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="TerminalDisplayItemType" nillable="true" type="TerminalDisplayItemType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TerminalDisplayItemType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Quantity" type="xs:int" />  <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Price" type="xs:double" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TerminalDebitConfigurationType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="AllowDebit" type="xs:boolean" />  <xs:element minOccurs="0" maxOccurs="1" name="CashbackAmounts" type="ArrayOfInt" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfInt">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="int" type="xs:int" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="EMVReceiptQueryType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="1" maxOccurs="1" name="TransactionReference" type="xs:string" />  </xs:sequence>  </xs:complexType>  </xs:schema> |

Response

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="Response\_v1" nillable="true" type="Response\_v1" />  <xs:complexType name="Response\_v1">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="PaymentResponses" type="ArrayOfPaymentResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="BatchResponse" type="BatchResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionStatusQueryResponses" type="ArrayOfTransactionStatusQueryResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="RecurringStatusQueryResponses" type="ArrayOfRecurringStatusQueryResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultStatusQueryResponse" type="VaultStatusQueryResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultResponse" type="VaultResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultAccountResponse" type="VaultAccountResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="AccountQueryResponse" type="AccountQueryResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TerminalItemListDisplayResponse" type="TerminalItemListDisplayResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="HealthcareSignatureResponse" type="HealthcareSignatureResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="EMVReceiptQueryResponse" type="EMVReceiptQueryResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfPaymentResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="PaymentResponseType" nillable="true" type="PaymentResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="PaymentResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultResponse" type="VaultResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="RecurringResponse" type="RecurringResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionResponse" type="TransactionResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="Customer" type="PersonType" />  <xs:element minOccurs="0" maxOccurs="1" name="ShippingRecipient" type="PersonType" />  <xs:element minOccurs="0" maxOccurs="1" name="Receipt" type="ReceiptType" />  <xs:element minOccurs="0" maxOccurs="1" name="EmvResponse" type="EmvResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="MonerisReceipt" type="MonerisReceiptType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="ResponseIndicator" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ResponseCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ResponseMessage" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="GUID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ExpirationDate" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Last4" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="PaymentDescription" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="PaymentTypeID" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="RecurringResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="RecurringID" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TransactionResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="AuthCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AVSResult" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CVVResult" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="VANReference" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Last4" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="PaymentDescription" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Amount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="PaymentTypeID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Reference1" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionDate" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AuxiliaryData" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="EntryMode" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TaxAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="ShippingAmount" type="xs:double" />  <xs:element minOccurs="0" maxOccurs="1" name="CustomerNumber" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ACHTransactionClass" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TransactionPaymentType" type="TransactionPaymentType" />  <xs:element minOccurs="1" maxOccurs="1" name="CashbackAmount" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="FSACard" type="xs:boolean" />  <xs:element minOccurs="0" maxOccurs="1" name="NetworkID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CardExpirationDate" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CurrencyCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SignatureImageData" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SignatureFormat" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:simpleType name="TransactionPaymentType">  <xs:restriction base="xs:string">  <xs:enumeration value="CREDITCARD" />  <xs:enumeration value="PURCHASECARD" />  <xs:enumeration value="VIRTUALCHECK" />  </xs:restriction>  </xs:simpleType>  <xs:complexType name="PersonType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="NameType" />  <xs:element minOccurs="0" maxOccurs="1" name="Address" type="AddressType" />  <xs:element minOccurs="0" maxOccurs="1" name="Company" type="CompanyType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="NameType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="FirstName" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MI" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="LastName" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="AddressType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="AddressLine1" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AddressLine2" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="City" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="State" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ZipCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Country" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="EmailAddress" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Telephone" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Fax" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="CompanyType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Address" type="AddressType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ReceiptType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="ReceiptID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ReceiptTypeID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Language" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ReceiptData" type="ArrayOfNameValueType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfNameValueType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="NameValueType" nillable="true" type="NameValueType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="NameValueType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="MonerisReceiptType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Account" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Amount" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AppLabel" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AppPreferredName" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ARQC" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="AuthCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="BalanceAmount" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CardNumber" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CardPlan" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CardType" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CashbackAmount" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="CvmIndicator" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="EmvAID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="InvoiceNumber" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="IsoCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantAddress" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantCity" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantName" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantState" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantZip" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="PanEntry" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="PartialAmount" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="RefNum" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="ResponseCode" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SupplementalData" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SupplementalData2" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TipAmount" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TotalAmount" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TransDate" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TransTime" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TransType" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TSI" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TvrARQC" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="TvrTCACC" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Type" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="EmvResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="EmvTags" type="ArrayOfEmvTagType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfEmvTagType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="EmvTagType" nillable="true" type="EmvTagType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="EmvTagType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Tag" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="BatchResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="BatchNumber" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="BatchReference" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="Net" type="xs:double" />  <xs:element minOccurs="1" maxOccurs="1" name="Count" type="xs:int" />  <xs:element minOccurs="1" maxOccurs="1" name="BatchPayment" type="BatchPaymentType" />  </xs:sequence>  </xs:complexType>  <xs:simpleType name="BatchPaymentType">  <xs:restriction base="xs:string">  <xs:enumeration value="CREDITCARD" />  <xs:enumeration value="PURCHASECARD" />  <xs:enumeration value="VIRTUALCHECK" />  </xs:restriction>  </xs:simpleType>  <xs:complexType name="ArrayOfTransactionStatusQueryResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="TransactionStatusQueryResponseType" nillable="true" type="TransactionStatusQueryResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TransactionStatusQueryResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultResponse" type="VaultResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="RecurringResponse" type="RecurringResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionResponse" type="TransactionResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionSettlementStatus" type="TransactionSettlementStatusType" />  <xs:element minOccurs="0" maxOccurs="1" name="Customer" type="PersonType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TransactionSettlementStatusType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="TransactionType" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SettlementType" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="SettlementDate" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="BatchReference" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfRecurringStatusQueryResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="RecurringStatusQueryResponseType" nillable="true" type="RecurringStatusQueryResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="RecurringStatusQueryResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Responses" type="ArrayOfResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionResponses" type="ArrayOfTransactionResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="TransactionSettlementStatuses" type="ArrayOfTransactionSettlementStatusType" />  <xs:element minOccurs="0" maxOccurs="1" name="Customers" type="ArrayOfPersonType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="ResponseType" nillable="true" type="ResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfTransactionResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="TransactionResponseType" nillable="true" type="TransactionResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfTransactionSettlementStatusType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="TransactionSettlementStatusType" nillable="true" type="TransactionSettlementStatusType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfPersonType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="PersonType" nillable="true" type="PersonType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultStatusQueryResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultResponse" type="VaultResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultAccountResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="VaultAccount" type="VaultAccountType" />  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="VaultAccountType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Company" type="CompanyType" />  <xs:element minOccurs="0" maxOccurs="1" name="Contact" type="PersonType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="MerchantType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="MerchantID" type="xs:string" />  <xs:element minOccurs="0" maxOccurs="1" name="MerchantKey" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="AccountQueryResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="Merchant" type="MerchantType" />  <xs:element minOccurs="0" maxOccurs="1" name="Company" type="CompanyType" />  <xs:element minOccurs="0" maxOccurs="1" name="Services" type="ArrayOfString" />  <xs:element minOccurs="1" maxOccurs="1" name="Active" type="xs:boolean" />  <xs:element minOccurs="0" maxOccurs="1" name="CreditCardTerminalConfigurations" type="ArrayOfGatewayTerminalConfiguration" />  <xs:element minOccurs="0" maxOccurs="1" name="VirtualCheckTerminalConfigurations" type="ArrayOfGatewayTerminalConfiguration" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfString">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="ArrayOfGatewayTerminalConfiguration">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="unbounded" name="GatewayTerminalConfiguration" nillable="true" type="GatewayTerminalConfiguration" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="GatewayTerminalConfiguration">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="TerminalNumber" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="FrontEndId" type="xs:int" />  <xs:element minOccurs="0" maxOccurs="1" name="Settings" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="TerminalItemListDisplayResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="HealthcareSignatureResponseType">  <xs:sequence>  <xs:element minOccurs="0" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="1" name="ImageData" type="xs:string" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="EMVReceiptQueryResponseType">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="Response" type="ResponseType" />  <xs:element minOccurs="0" maxOccurs="unbounded" name="EMVReceipts" type="EMVReceiptData" />  </xs:sequence>  </xs:complexType>  <xs:complexType name="EMVReceiptData">  <xs:sequence>  <xs:element minOccurs="1" maxOccurs="1" name="MerchantName" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TransactionDateTime" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TransactionID" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TransactionType" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="CardNumber" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="UTI" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="EntryMode" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="CVM" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="AuthMessage" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="AuthCode" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TaxAmount" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="ShippingAmount" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TotalAmount" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="AID" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TVR" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="IAD" type="xs:string" />  <xs:element minOccurs="1" maxOccurs="1" name="TSI" type="xs:string" />  </xs:sequence>  </xs:complexType>  </xs:schema> |

XML Reserved Characters

In XML, some characters are reserved for internal use and you must replace them by entity references when they are used in data submitted to the Sage Exchange. The following table shows the characters that must be replaced by their entity references.

Methods

|  |  |
| --- | --- |
| Character | Entity reference |
| > | &gt; |
| < | &lt; |
| & | &amp; |
| % | &#37; |

Message formats

The following illustrations provide examples of how the request message components are structured. All requests are framed within a Request\_v1 root and must include an Application element. Optional XML elements can be omitted from the request message when not used. Some elements are arrays; in cases where more than one element is present in the array processing will occur in FIFO. XML reserved characters must be observed and encoded appropriately.

Sale

Sale is a transaction type in which the transaction is authorized and captured in one step. Sale is used only to process purchases of goods or services that do not require physical shipment or "soft" goods that are delivered electronically.

XML sale request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

XML sale request without user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>e856a127-8527-431c-807f-6efacd8bdf83</TransactionID>  <TransactionType>01</TransactionType>  <Reference1>INV# 451777674</Reference1>  <Amount>2152.92</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>John</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>john.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  <VaultStorage>  <GUID>dd83d7559a274fb2b66e774a4febced7</GUID>  <Service>RETRIEVE</Service>  </VaultStorage>  </PaymentType>  </Payments>  </Request\_v1> |

XML sale request and vault storage with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>e856a127-8527-431c-807f-6efacd8bdf83</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 451777674</Reference1>  <Amount>2152.92</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>John</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>john.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  <VaultStorage>  <Service>CREATE</Service>  </VaultStorage>  </PaymentType>  </Payments>  </Request\_v1> |

Authorization

Authorization is a transaction type in which an account is verified to be valid and has not reached its limit. The total amount of the transaction is reserved against the account balance. Authorizations are used if goods are to be physically shipped or in other cases for which the merchant must first verify whether the order can be fulfilled. An approved Authorization is followed by a Capture, which prepares it for settlement.

XML authorization request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>12</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

XML authorization request without user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>e856a127-8527-431c-807f-6efacd8bdf83</TransactionID>  <TransactionType>02</TransactionType>  <Reference1>INV# 451777674</Reference1>  <Amount>2152.92</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>John</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>john.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  <VaultStorage>  <GUID>dd83d7559a274fb2b66e774a4febced7</GUID>  <Service>RETRIEVE</Service>  </VaultStorage>  </PaymentType>  </Payments>  </Request\_v1> |

Capture

Capture is a transaction type that puts an Authorization transaction into a Captured state for settlement. In the case of partial shipments, the Capture amount may be less than the Authorization amount. Captures can be initiated only after the purchased goods have been shipped.

XML capture request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>0405aa29-9be2-4c46-b8b0-b103e25a39b6</TransactionID>  <TransactionType>13</TransactionType>  <Amount>4577.52</Amount>  <VANReference>CBK9A0j650</VANReference>  </TransactionBase>  </PaymentType>  </Payments>  </Request\_v1> |

XML capture request without user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>0405aa29-9be2-4c46-b8b0-b103e25a39b6</TransactionID>  <TransactionType>03</TransactionType>  <Amount>4577.52</Amount>  <VANReference>CBK9A0j650</VANReference>  </TransactionBase>  </PaymentType>  </Payments>  </Request\_v1> |

Force

Force is a transaction type used to force a transaction into settlement in cases where a Sale or Authorization transaction cannot be processed and the Authorization Code is obtained from an outside source.

XML force request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>15</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  <AuthCode>123456</AuthCode>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

XML force request without user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>e856a127-8527-431c-807f-6efacd8bdf83</TransactionID>  <TransactionType>05</TransactionType>  <Reference1>INV# 451777674</Reference1>  <Amount>2152.92</Amount>  <AuthCode>123456</AuthCode>  </TransactionBase>  <Customer>  <Name>  <FirstName>John</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>john.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  <VaultStorage>  <GUID>dd83d7559a274fb2b66e774a4febced7</GUID>  <Service>RETRIEVE</Service>  </VaultStorage>  </PaymentType>  </Payments>  </Request\_v1> |

Level 2

Level2 data is additional transaction data required for Level II commercial card qualification for credit card transactions.

XML Level 2 sale request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  <Level2>  <CustomerNumber>123456789012</CustomerNumber>  <TaxAmount>92.59</TaxAmount>  </Level2>  </PaymentType>  </Payments>  </Request\_v1> |

Void

Void is a transaction type that cancels a transaction that has not yet been settled.

XML void request without user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>782ad8d0-0dd2-4763-8d64-cc9fddfad441</TransactionID>  <TransactionType>04</TransactionType>  <VANReference>ABL9LKQaI0</VANReference>  </TransactionBase>  </PaymentType>  </Payments>  </Request\_v1> |

Credit

Credit is a transaction type that transfers funds to the account, rather than from the account. This transaction type is typically used to refund money for a transaction that was previously settled.

XML credit request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>0405aa29-9be2-4c46-b8b0-b103e25a39b6</TransactionID>  <TransactionType>16</TransactionType>  <Amount>4577.52</Amount>  <VANReference>CBK9A0j650</VANReference>  </TransactionBase>  </PaymentType>  </Payments>  </Request\_v1> |

XML credit request without user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>0405aa29-9be2-4c46-b8b0-b103e25a39b6</TransactionID>  <TransactionType>06</TransactionType>  <Amount>4577.52</Amount>  <VANReference>CBK9A0j650</VANReference>  </TransactionBase>  </PaymentType>  </Payments>  </Request\_v1> |

Single payment response

Below is a typical response for a single payment.

XML payment response

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Response\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <PaymentResponses>  <PaymentResponseType>  <Response>  <ResponseIndicator>A</ResponseIndicator>  <ResponseCode>024702</ResponseCode>  <ResponseMessage>APPROVED </ResponseMessage>  </Response>  <TransactionResponse>  <AuthCode>024702</AuthCode>  <AVSResult />  <CVVResult>N</CVVResult>  <VANReference>GAUD9m1Bj0</VANReference>  <TransactionID>70d0d88f-bc2d-4c61-a6ad-843dab2cb649</TransactionID>  <Last4>XXXXXXXXXXXX1111</Last4>  <PaymentDescription>411111XXXXXX1111</PaymentDescription>  <Amount>19.99</Amount>  <PaymentTypeID>4</PaymentTypeID>  <Reference1>PO 123456</Reference1>  <TransactionDate>10/30/2013 9:47:00 AM</TransactionDate>  <EntryMode>K</EntryMode>  <TaxAmount>0</TaxAmount>  <ShippingAmount>0</ShippingAmount>  <TransactionPaymentType>CREDITCARD</TransactionPaymentType>  <CashbackAmount>0</CashbackAmount>  <FSACard>false</FSACard>  </TransactionResponse>  <Customer>  <Name>  <FirstName>John A Doe</FirstName>  <MI />  <LastName />  </Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2>Apt #2</AddressLine2>  <City>Some City</City>  <State>Alabama</State>  <ZipCode>12345</ZipCode>  <Country>United States</Country>  <EmailAddress>john.doe@domain.com</EmailAddress>  <Telephone>1234567891</Telephone>  <Fax>1234567890</Fax>  </Address>  <Company>  <Name>John's Company</Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2>Apt #2</AddressLine2>  <City>Some City</City>  <State>Some State</State>  <ZipCode>12345</ZipCode>  <Country>Some Country</Country>  <EmailAddress>john.doe@domain.com</EmailAddress>  <Telephone>1234567891</Telephone>  <Fax>1234567890</Fax>  </Address>  </Company>  </Customer>  <ShippingRecipient>  <Name>  <FirstName>John</FirstName>  <MI>A</MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2>Apt #2</AddressLine2>  <City>Some City</City>  <State>Some State</State>  <ZipCode>12345</ZipCode>  <Country>Some Country</Country>  <EmailAddress>john.doe@domain.com</EmailAddress>  <Telephone>1234567891</Telephone>  <Fax>1234567890</Fax>  </Address>  <Company>  <Name>John's Company</Name>  <Address>  <AddressLine1>12345 Street</AddressLine1>  <AddressLine2>Apt #2</AddressLine2>  <City>Some City</City>  <State>Some State</State>  <ZipCode>12345</ZipCode>  <Country>Some Country</Country>  <EmailAddress>john.doe@domain.com</EmailAddress>  <Telephone>1234567891</Telephone>  <Fax>1234567890</Fax>  </Address>  </Company>  </ShippingRecipient>  </PaymentResponseType>  </PaymentResponses>  </Response\_v1> |

Batch inquiry

Batch inquiry is used to obtain the transaction net and transaction count of the current open batch awaiting settlement.

XML batch inquiry request

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Batch>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <Net>-1</Net>  <Count>-1</Count>  <BatchPayment>CREDITCARD</BatchPayment>  </Batch>  </Request\_v1> |

Batch close

Batch Close is used to settle transactions in the current open batch awaiting settlement. Sales, Captures, Forces, and Credit transactions qualify for settlement.

XML batch close request with net and count verification

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Batch>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <Net>2561.23</Net>  <Count>5</Count>  <BatchPayment>CREDITCARD</BatchPayment>  </Batch>  </Request\_v1> |

XML batch close request without net and count verification

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Batch>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <Net>0</Net>  <Count>0</Count>  <BatchPayment>CREDITCARD</BatchPayment>  </Batch>  </Request\_v1> |

Vault operation

Vault Operation is used to capture sensitive card holder data and insert or update the storage in the Sage Payment Solutions Vault. No payment is processed.

XML vault operation request for creating a storage record

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <VaultOperation>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <VaultStorage>  <Service>CREATE</Service>  </VaultStorage>  <VaultID>2341234-12431243-2341235</VaultID>  </VaultOperation>  </Request\_v1> |

XML vault operation request for updating a storage record

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <VaultOperation>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <VaultStorage>  <Service>UPDATE</Service>  <GUID>sfdas-ee3u38d-dagdi3-efad83</GUID>  </VaultStorage>  <VaultID>2341234-12431243-2341235</VaultID>  </VaultOperation>  </Request\_v1> |

Transaction status query

Transaction Status Query is used to get the status of a transaction processed through the Sage Exchange by using the user defined Transaction ID provided during a previous Payment. In the event of a communication failure in which the response was not received this can be used to determine if the platform received and processed the transaction or if the transaction needs to be run again. This can be used to prevent duplicated transactions. If post transaction logic is required this can also provide additional information such as the current state of the transaction (settled/expired/voided).

XML transaction status query

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <TransactionStatusQueries>  <TransactionStatusQueryType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionID>sdfasdf089273412903479a87sa</TransactionID>  </TransactionStatusQueryType>  </TransactionStatusQueries>  </Request\_v1> |

Vault status query

Vault Status Query is used to get the status of a Vault operation processed through the Sage Exchange by using the user defined Vault ID provided during a previous Vault operation. In the event of a communication failure in which the response was not received this can be used to determine if the platform received and processed the operation or if the operation needs to be run again.

XML vault status query

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <VaultStatusQuery>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <VaultID>sdfasdf089273412903479a87sa</VaultID>  </VaultStatusQuery>  </Request\_v1> |

Multi-payment processing

The Sage Exchange application supports processing multiple payments in a single XML request. The request can combine both UI and Non-UI payments. If any payment contains a UI transaction type then the Multi-Payment UI will be shown.

XML multi-payment request with user interface

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>4fa9747c-13a2-46af-970f-f8a92f5d4f61</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 7563456</Reference1>  <Amount>50.50</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>John</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>4567 Street</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>[john.doe@sagepayments.com</EmailAddress](mailto:john.doe@sagepayments.com%3c/EmailAddress)>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

Split payment processing

The Sage Exchange application supports splitting a single payment across multiple merchant accounts. The total amount of the payment is the combine Amount of all the payments in the array. The first account in the payments array is considered the primary merchant account. The request can combine both UI and Non-UI payments. If any payment contains a UI transaction type then the Split Payment UI will be shown.

XML split payment request with user interface

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <IsSplitPayment>true</IsSplitPayment>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>12</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1.00</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  <PaymentType>  <Merchant>  <MerchantID>999999999990</MerchantID>  <MerchantKey>D8H8M8F6K7A7</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>12</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1.00</Amount>  </TransactionBase>  <Customer>  <Name>  <FirstName>John</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

Terminal item list

The Sage Exchange application supports hardware terminals that can display a listing of items being purchased. The follow request can be used to display a list of items on these support terminals.

XML terminal items list request

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <TerminalItemListDisplay>  <TerminalDisplayItems>  <TerminalDisplayItemType>  <Quantity>5</Quantity>  <Description>Item 1</Description>  <Price>1.00</Price>  </TerminalDisplayItemType>  <TerminalDisplayItemType>  <Quantity>10</Quantity>  <Description>Item 2</Description>  <Price>2.12</Price>  </TerminalDisplayItemType>  </TerminalDisplayItems>  </TerminalItemListDisplay>  </Request\_v1> |

Healthcare signature

The Sage Exchange application supports hardware terminals that can capture an image of a signature. Signatures that are captured are linked to a previously process transaction via the VANReference.

XML healthcare signature request

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <HealthcareSignature>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <VANReference>A123456789</VANReference>  <SignatureAttributes>  <SignatureAttributeType>  <Name>COUNSELING</Name>  <DisplayText>I was offered couseling</DisplayText>  </SignatureAttributeType>  <SignatureAttributeType>  <Name>NARCOTICS</Name>  <DisplayText>I acknowledge receipt of the listed prescriptions</DisplayText>  </SignatureAttributeType>  <SignatureAttributeType>  <Name>EZCAP</Name>  <DisplayText>I acknowledge Ez Open Caps</DisplayText>  </SignatureAttributeType>  <SignatureAttributeType>  <Name>PRIVACY</Name>  <DisplayText>I acknowledge the Notice of Privacy Practices</DisplayText>  </SignatureAttributeType>  </SignatureAttributes>  <ItemsDisplayText>I acknowledge receipt of the listed prescriptions</ItemsDisplayText>  <Items>  <string>RX12345</string>  <string>RX345</string>  </Items>  </HealthcareSignature>  </Request\_v1> |

XML healthcare signature response

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Response\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <HealthcareSignatureResponse>  <Response>  <ResponseIndicator>A</ResponseIndicator>  <ResponseMessage>ACCEPTED</ResponseMessage>  </Response>  <ImageData></ImageData>  </HealthcareSignatureResponse>  </Response\_v1> |

Tender type

It is possible to explicitly request the tender type to be used for a given transaction. This is used when the calling application has already determined the method of payment. This is an example of a Sale transaction requesting credit/debit card processing.

XML credit card sale request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  <TenderType>CREDITCARD</TenderType>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

XML cash sale request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  <TenderType>CASH</TenderType>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  </Request\_v1> |

EMV Receipt Request

There are two ways to generate EMV receipt:

1. Run a successful EMV transaction. EMV receipt will be delivered as part of the response XML inside each PaymentResponseType element.

2. Request an already existing EMV receipt by providing the merchant info and corresponding transaction (i.e. VAN) reference.

XML EMV Receipt Request Sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID></ApplicationID>  <LanguageID></LanguageID>  </Application>  <EMVReceiptQuery>  <Merchant>  <MerchantID></MerchantID>  <MerchantKey></MerchantKey>  </Merchant>  <TransactionReference></TransactionReference>  </EMVReceiptQuery>  </Request\_v1> |

Terminal debit configuration

Sage Exchange supports hardware terminals that can process debit (PIN Entry) and cash back. By default Sage Exchange will process debit (PIN Entry) on terminals and cards that support it. If cash back is required additional amounts need to be provided. The following request is a Sale with the additional terminal debit configuration.

XML Credit Card Sale Request with User Interface and Terminal Debit Configuration Sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>1892.59</Amount>  <TenderType>CREDITCARD</TenderType>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  </PaymentType>  </Payments>  <TerminalDebitConfiguration>  <AllowDebit>true</AllowDebit>  <CashbackAmounts>  <int>10</int>  <int>20</int>  <int>30</int>  <int>40</int>  </CashbackAmounts>  </TerminalDebitConfiguration>  </Request\_v1> |

Healthcare

Healthcare data is additional transaction data required for Healthcare reporting.

XML healthcare sale request with user interface sample

|  |
| --- |
| <?xml version="1.0" encoding="utf-16"?>  <Request\_v1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <Application>  <ApplicationID>DEMO</ApplicationID>  <LanguageID>EN</LanguageID>  </Application>  <Payments>  <PaymentType>  <Merchant>  <MerchantID>999999999997</MerchantID>  <MerchantKey>K3QD6YWyhfD</MerchantKey>  </Merchant>  <TransactionBase>  <TransactionID>5ea9747c-12a4-46af-970f-f8a92f6d4f65</TransactionID>  <TransactionType>11</TransactionType>  <Reference1>INV# 886478943</Reference1>  <Amount>12.50</Amount>  <TenderType>CREDITCARD</TenderType>  </TransactionBase>  <Customer>  <Name>  <FirstName>Jane</FirstName>  <MI> </MI>  <LastName>Doe</LastName>  </Name>  <Address>  <AddressLine1>67890 Road</AddressLine1>  <AddressLine2></AddressLine2>  <City>South Padre Island</City>  <State>Texas</State>  <ZipCode>78597</ZipCode>  <Country>USA</Country>  <EmailAddress>jane.doe@sagepayments.com</EmailAddress>  <Telephone></Telephone>  <Fax></Fax>  </Address>  </Customer>  <Healthcare>  <HealthcareAmount>12.50</HealthcareAmount>  <ClinicAmount>0.00</ClinicAmount>  <DentalAmount>0.00</DentalAmount>  <VisionAmount>0.00</VisionAmount>  <PerscriptionAmount>0.00</PerscriptionAmount>  <IIASVerification>0</IIASVerification>  <HealthcareItems>  <HealthcareItemType>  <Quantity>2</Quantity>  <Description>My Item</Description>  <Price>6.25</Price>  </HealthcareItemType>  </HealthcareItems>  </Healthcare>  </PaymentType>  </Payments>  </Request\_v1> |

Request field definitions

Request\_V1 Type element - required

<Request\_V1>

<Application></Application>

<Payments></Payments>

<HealthcareSignature></HealthcareSignature>

<IsSplitPayment></IsSplitPayment>

</Request\_V1>

The Request\_V1 element is the root. It contains the Application, Payments, and IsSplitPayment elements.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Application | ApplicationType | Yes | Identifies the calling application, version, and its certification as a valid integrated solution. |
| Payments | Array of PaymementType | No | An array of payments to be processed. |
| HealthcareSignature | HealthcareSignatureType | No | Used for capturing a signature for Healthcare transactions. \*Only supported on specific hardware terminals. |
| IsSplitPayment | boolean | No | Indicates if the array of payments is to be treated as split payments or multi payments. If omitted the default is false. |

ApplicationType element - required

<Application>

<ApplicationID></ApplicationID>

<LanguageID></LanguageID>

<ClientID></ClientID>

<ClientKey></ClientKey>

</Application>

The Application element contains the ApplicationID, LanguageID, ClientID, and ClientKey elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| ApplicationID | string | 50 | Yes | Identifies the calling application, version, and its certification as a valid integrated solution. The value is obtained from Sage Payment Solutions through a registration/certification process. |
| LanguageID | string | 10 | No | Specifies the language to be used when displaying the user interface, the default is “en-US” for English United States. The values are derived from the lower case 2 letter language code from ISO 699-1 and the two letter upper case from ISO 3166. Example:  “fr-CA” = French Canadian |
| ClientID | string | 1-50 | Required for EMV. | A Sage-provided value that identifies the calling application during EMV transactions. |
| ClientKey | string | 1-50 | Required for EMV. | A Sage-provided value that identifies the calling application during EMV transactions. |

Payments element - optional

<Payments>

<PaymentType></PaymentType>

</Payments>

The Payments element can contain one or more PaymentType elements. When more than one payment is required to be processed by the calling application multiple PaymentType elements are used. When only a single payment is required only a single PaymentType element is used. At least one PaymentType element is required in a payment request message.

Batch element - optional

<Batch>

<BatchType></BatchType>

</Batch>

The Batch element can contain one BatchType element.

BatchType element

<BatchType>

<Merchant></Merchant>

<Net></Net>

<Count></Count>

<BatchPayment></BatchPayment>

</BatchType>

The BatchType element refers to a single payment and must contain a Merchant and BatchPayment element and may contain the optional Net and Count elements.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Merchant | MerchantType | Yes | Contains the merchant account elements related to processing a batch. |
| Net | decimal | No | The transaction total net amount of the batch being settled |
| Count | integer | No | The transaction count of the batch being settled |
| BatchPayment | BatchPaymentType | Yes | The payment type of batch being settled.  • CREDITCARD = Standard Credit Card transactions  • PURCHASECARD = Level III qualified transactions  • VIRTUALCHECK = ACH transactions |

HealthcareSignatureType element - optional

The HealthcareSignature element contains elements required for a hardware terminal to prompt, capture, and store a healthcare signature with a transaction.

<HealthcareSignature>

<Merchant></Merchant>

<VANReference> </VANReference>

<SignatureAttributes></SignatureAttributes>

<ItemsDisplayText></ItemsDisplayText>

<Items></Items>

</HealthcareSignature>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Merchant | MerchantType |  | Yes | Contains the merchant account elements related to processing a payment. |
| VANReference | string | 10 | Yes | The reference of the transaction to store the image to. |
| ItemsDisplayText | string |  | No | The text displayed above the items being signed for on the terminal. |
| Items | complexType |  | No | An array of strings, each describing an item being signed for. If this is omitted the terminal will skip displaying the list of items. |

PaymentType element

<PaymentType>

<Merchant></Merchant>

<TransactionBase></TransactionBase>

<Customer></Customer>

<ShippingRecipient></ShippingRecipient>

<Level2></Level2>

<Level3></Level3>

<VaultStorage></VaultStorage>

<Recurring></Recurring>

<Postback></Postback>

<Healthcare></Healthcare>

</PaymentType>

The PaymentType element refers to a single payment and must contain a Merchant and TransactionBase element and may contain the optional Customer, ShippingRecipient, Level2, Level3, VaultStorage, Recurring, Healthcare elements.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Merchant | MerchantType | Yes | Contains the merchant account elements related to processing a payment. |
| TransactionBase | TransactionBaseType | Yes | Contains the transaction elements related to processing a payment. |
| Customer | PersonType | No\* | Contains the elements related to the person making the payment. |
| ShippingRecipient | PersonType | No | Contains the elements related to the person receiving the goods/services related to the payment. |
| Level2 | Level2Type | No | Contains the additional transaction elements related to qualifying for Level II. The Level2 and Level3 elements are mutually exclusive and should not be used together in a payment. |
| Level3 | Level3Type | No | Contains the additional transaction elements related to qualifying for Level III. The Level2 and Level3 elements are mutually exclusive and should not be used together in a payment. |
| VaultStorage | VaultStorageType | No\* | Contains the elements related to Storing/Retrieving/Updating data in the Vault service. |
| Recurring | RecurringType | No | Contains the elements needed to schedule a payment in the recurring system. |
| Postback | PostbackType | No | Contains the elements related to POSTing transaction response data to a publically available user defined URL. |
| Healthcare | HealthcareType | No | Contains the additional Healthcare elements related to processing FSA / Healthcare transactions. |

\*Element is required for a Sale, Authorization, or Force without a user interface.

MerchantType element

<Merchant>

<MerchantID></MerchantID>

<MerchantKey></MerchantKey>

</Merchant>

The Merchant element contains the required MerchantID and MerchantKey elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| MerchantID | string | 12 | Yes | Identifies the merchant account on the VAN |
| MerchantKey | string | 12 | Yes | Identifies the merchant account on the VAN |

TransactionBaseType element

<TransactionBase>

<TransactionID></TransactionID>

<TransactionType></TransactionType>

<Reference1></Reference1>

<Amount></Amount>

<AuthCode></AuthCode>

<VANReference></VANReference>

</TransactionBase>

The TransactionBase element contains the TransactionID, TransactionType, Reference1, Amount, AuthCode, TenderType, CustomerType, and VANReference elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| TransactionID | string | 32 | Yes | Identifies the payment for status inquires. In the event the communication is interrupted a payment response can be queried later using this element. |
| TransactionType | integer | 2 | Yes | No user interface:  01 = Sale  02 = Authorization  03 = Capture  04 = Void  05 = Force  06 = Credit  07 = Credit without Reference  User interface:  11 = Sale  12 = Authorization  13 = Capture  15 = Force  16 = Credit  17 = Credit without Reference |
| Reference1 | string | 50 | No | User defined field, like Invoice Number, Purchase Order Number, etc… |
| Amount | decimal |  | No | Amount of the payment |
| AuthCode | string | 6 | No\* | Authorization Code for Force transactions |
| VANReference | string | 10 | No\*\* | VAN unique transaction identifier used with Credit, Capture, or Void transactions |
| TenderType | TransactionTenderType |  | No | Used to control the payment type used for payment processing.  CREDITCARD = Debit / Credit Card payment  CASH = Cash or a miscellaneous payment  VIRTUALCHECK = Ach payment |
| CustomerType | CustomerType |  | No | The type of customer processing the payment, the default is Consumer.  Business = For business to business payments  Consumer = For consumer to business payments  Government = For government to business payments |

\*Element is required for a Force transaction

\*\*Element is required for a Credit, Capture, or Void transaction

Customer element: PersonType element

<Customer>

<Name></Name>

<Address></Address>

<Company></Company>

</Customer>

The Customer element contains the Name, Address, and Company elements for the person making the payment.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Name | NameType | No | Contains elements related to the Customer’s name. |
| Address | AddressType | No\* | Contains elements related to the billing address, used during Address Verification Service for manually keyed transactions. |
| Company | CompanyType | No | Contains elements related to the Customer’s company. |

\*Elements used for Credit Card AVS processing when the payment is manually keyed.

ShippingRecipient element: PersonType element

<ShippingRecipient>

<Name></Name>

<Address></Address>

<Company></Company>

</ShippingRecipient>

The ShippingRecipient element contains the Name, Address, and Company elements for the person receiving the goods/service.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Name | NameType | No | Contains elements related to the Recipient’s name. |
| Address | AddressType | No\* | Contains elements related to the shipping address. |
| Company | CompanyType | No | Contains elements related to the Recipient’s company. |

\*Elements used for Credit Card AVS processing when the payment is manually keyed.

Level2Type element

<Level2>

<CustomerNumber></CustomerNumber>

<TaxAmount></TaxAmount>

</Level2>

The Level2 element contains the CustomerNumber and TaxAmount elements used to qualify a Purchase Card payment for Level II.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| CustomerNumber | string | 17 | Yes | User defined Customer Number |
| TaxAmount | decimal |  | Yes | The tax amount being charged |

Level3Type element

<Level3>

<Level2></Level2>

<ShippingAmount></ShippingAmount>

<DestinationZipCode></DestinationZipCode>

<DestinationCountryCode></DestinationCountryCode>

<VATNumber></VATNumber>

<DiscountAmount></DiscountAmount>

<DutyAmount></DutyAmount>

<NationalTaxAmount></NationalTaxAmount>

<VATInvoiceNumber></VATInvoiceNumber>

<VATTaxAmount></VATTaxAmount>

<VATTaxRate></VATTaxRate>

<LineItems></LineItems>

</Level3>

The Level3 element contains the optional CustomerNumber, TaxAmount, DestinationZipCode, NationalTax, VATNumber, DiscountAmount, DutyAmount, VATInvoiceNumber, VATTaxAmount, VATTaxRate, DestinationCountryCode, and LineItems elements used to qualify a Purchase Card payment for Level III.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Level2 | string | 17 | Yes | Contains the Level2 elements CustomerNumber and TaxAmount |
| ShippingAmount | decimal |  | Yes | Shipping amount charged to the transaction |
| DestinationZipCode | string | 9 | Yes | Postal zip code where the goods/services are shipped |
| VATNumber | string | 13 | Yes | Customer’s Value Added Tax Number |
| DiscountAmount | decimal |  | Yes | Discount amount applied to the transaction |
| DutyAmount | decimal |  | Yes | Duty amount |
| VATInvoiceNumber | string | 15 | Yes | Value Added Tax invoice number |
| VATTaxAmount | decimal |  | Yes | Value Added Tax amount |
| VATTaxRate | decimal |  | Yes | Value Added Tax rate |
| DestinationCountryCode | integer |  | Yes | 3 digit ISO Country Code where the goods/services are shipped |
| LineItems | Array of Level3LineItemType |  | No | Contains zero or more Level3LineItemType elements |
| NationalTaxAmount | decimal |  | Yes | National Tax amount applied to the transaction |

Level3LineItemType element

<Level3LineItemType>

<CommodityCode></CommodityCode>

<Description></Description>

<ProductCode></ProductCode>

<Quantity></Quantity>

<UnitOfMeasure></UnitOfMeasure>

<UnitCost></UnitCost>

<TaxAmount></TaxAmount>

<TaxRate></TaxRate>

<DiscountAmount></DiscountAmount>

<AlternateTaxIdentifier></AlternateTaxIdentifier>

<TaxTypeApplied></TaxTypeApplied>

<DiscountIndicator></DiscountIndicator>

<NetGrossIndicator></NetGrossIndicator>

<ExtendedItemAmount></ExtendedItemAmount>

<DebitCreditIndicator></DebitCreditIndicator>

</Level3LineItemType>

The Level3LineItemType element contains the CommodityCode, Description, ProductCode, Quantity, UnitOfMeasure, UnitCost, TaxAmount, TaxRate, DiscountAmount, AlternateTaxIndentifier, TaxTypeApplied, DiscountIndicator, NetGrossIndicator, ExtendedItemAmount, DebitCreditIndicator, and TotalAmount elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| CommodityCode | string | 12 | Yes | Commodity code that applies to the item |
| Description | string | 35 | Yes | A brief description of the item |
| ProductCode | string | 12 | Yes | Product code that applies to the item |
| Quantity | integer |  | Yes | Quantity of item(s) purchased |
| UnitOfMeasure | string | 12 | Yes | Units of measure of the item(s) purchased |
| UnitCost | decimal |  | Yes | Cost of the item purchased |
| TaxAmount | decimal |  | Yes | The tax amount for the item |
| TaxRate | decimal | 15 | Yes | The tax rate for the item |
| DiscountAmount | decimal |  | Yes | Discount amount applied to item |
| AlerternateTaxIdentifier | string |  | Yes | Value Added Tax rate |
| TaxTypeApplied | string | 4 | Yes |  |
| DiscountIndiciator | string | 1 | Yes |  |
| ExtendedItemAmount | decimal |  | Yes | The total amount of the individual item. |
| = ( ItemCost X Quantity ) – ( DiscountAmount x Quantity ) |  |  |  |  |
| NetGrossIndiciator | string | 1 |  |  |

VaultStorageType element

<VaultStorage>

<Service></Service>

<GUID></GUID>

</VaultStorage>

The Vault element contains the Service, and GUID elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Service | VaultServiceType |  | Yes | Used to indicate the Vault operation.  RETRIEVE = Pull data from Vault for processing  UPDATE = Update data in Vault with new data captured  CREATE = Insert data in the Vault with data captured |
| GUID | string | 36 | No\* | The Vault GUID referencing a previous payment account captured in the Vault. Payment information will not need to be captured and instead it is retrieved from the Vault. |

\*Required for retrieval and update operations.

## RecurringType element

<Recurring>

<Schedule></Schedule>

<Interval></Interval>

<DayOfMonth></DayOfMonth>

<StartDate></StartDate>

<Amount></Amount>

<TimesToProcess></TimesToProcess>

<NonBusinessDay></NonBusinessDay>

</Recurring>

The Recurring element contains the ScheduleType, ScheduleInterval, DayOfMonth, StartDate, Amount, TimesToProcess, and NonBusinessDay elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Schedule | ScheduleType |  | Yes | The schedule type for the recurring payment  • DAILY = Schedule will be based on day  • MONTHLY = Schedule will be based on month |
| Interval | integer |  | No | The interval between processing.  Example: Every Other Month  Schedule = MONTHLY  Interval = 2 |
| DayOfMonth | integer |  | No | For monthly based schedule this is the day of the month to process the payment |
| StartDate | string | 10 | No | The date the payment will start processing, format MM/DD/YYYY |
| Amount | decimal |  | No | The amount of the payment, if different than the original payment amount |
| TimesToProcess | integer |  | No | The number of payments to process (-1 = Indefinite) |
| NonBusinessDays | NonBusinessDayType |  | No | • THATDAY = Payment is process on the non business day  • BEFORE = Payment is processed before the non business day  • AFTER = payment is processed after the non business day |

PostbackType element

<Postback>

<HttpsUrl></HttpsUrl>

</Postback>

The PostbackType element refers to a URL in which transaction response data should be sent after processing and must contain an HttpsUrl element.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| HttpsUrl | string | 1024 | Yes | The absolute URL to POST transaction response data back to. The communication requires SSL. This is to be used in integrations in which response handling is disconnected or in which response data should be delivered to a remote server. |

Response field definitions

PaymentResponseType element - required

<PaymentResponseType>

<Response></Respopnse>

<VaultResponse></VaultResponse>

<RecurringResponse></RecurringResponse>

<Customer></Customer>

<TransactionResponse></TransactionResponse>

<EmvResponse></EmvResponse>

</PaymentResponseType>

The PaymentResponseType element contains the response elements related to a payment.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Response | ResponseType | Yes | Contains the response elements related to the Gateway request |
| VaultResponse | VaultResponseType | No | Contains the response elements related to the Vault operation |
| RecurringResponse | RecurringResponseType | No | Contains the response elements related to the transaction enlistment into the recurring system |
| TransactionResponse | TransactionResponseType | Yes | Contains the response elements realted to processing a transaction on the Gateway |
| Customer | PersonType | Yes | Contains the billing/customer elements used for the payments |
| EmvResponse | EmvResponseType | No | Contains response data specific to EMV transactions. |
| MonerisReceipt | MonerisReceiptType | No | Contains response data specific to Moneris transactions. |

ResponseType element - required

<ResponseType>

<ResponseIndicator></RespopnseIndicator>

<ResponseCode></ResponseCode>

<ResponseMessage></ResponseMessage>

</ResponseType>

The ResponseType element contains the the Gateway response elements related to a Gateway request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| ResponseIndicator | string | 1 | Yes | The Gateway/Vault response indicator  A=Approved E=Declined X=Error  I = Batch Inquiry ( only returned on batch inquiry requests )  When returned during a Payment request this field is used to determine the status of a Payment ( Approved/Declined/Error) |
| ResponseCode | string | 6 | Yes | The Gateway/Vault response code, when the indicator is a E or X this code and be used to determine the cause |
| ResponseMessage | string | 32 | Yes | The Gateway/Vault response text |

VaultResponseType element

<VaultResponseType>

<Response></Response>

<GUID></GUID>

<ExpirationDate></ExpirationDate>

<Last4></Last4>

<PaymentDescription></PaymentDescription>

<PaymentTypeID></PaymentTypeID>

</VaultResponseType>

The VaultResponseType element contains the the response elements related to a Vault operation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Response | ResponseType |  | Yes | Contains the response elements related to the Vault request |
| GUID | string | 36 | Yes | The Vault GUID used to reference the card data captured |
| ExpirationDate | string | 4 | Yes | The date the account data will expire, format MMYY  \*This field is only applicable to responses with PaymentTypeID ( 3,4,5,6,7,D,O ) |
| Last4 | string |  | Yes | A masked respresentation of the account data showing only the last four digits |
| PaymentDescription | string |  | Yes | The description of the account data.  Credit Card payments will be the First 6 digits + masked digits + Last 4 digits ACH payments will be the routinng number + space + masked account digsts + Last 4 account digits |
| PaymentTypeID | string | 1 | Yes | The payment type identifier  3=American Express  4=Visa  5=MasterCard  6=Discover  7=JCB  D=Debit Card O=Other C=ACH |

RecurringResponseType element

<RecurringResponseType>

<RecurringID></RecurringID>

</RecurringResponseType>

The RecurringResponseType element contains the the recurring indentifier element to reference the recurring transaction.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| RecurringID | string | Yes | The recurring record identifier for the recurring transaction enlised in the recurring system |

TransactionResponseType element

<TransactionResponseType>

<AuthCode></AuthCode>

<AVSResult></AVSResult>

<CVVResult></CVVResult>

<VANReferene></VANReference>

<TransactionID></TransactionID>

<Last4></Last4>

<PaymentDescription></PaymentDescription>

<Amount></Amount>

<PaymentTypeID></PaymentTypeID>

<EntryMode></EntryMode>

<NetworkID></NetworkID>

<CardExpirationDate></CardExpirationDate>

<CurrencyCode></CurrencyCode>

<SignatureImageData></SignatureImageData>

<SignatureFormat></SignatureFormat>

</TransactionResponseType>

The TransactionResponseType element contains the the response elements related to processing a transaction.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| AuthCode | string | 6 | The authorization code for the approved payment |
| AVSResult | string | 1 | The AVS result for manually keyed transactions.  This field is for infromation purposes and is not to be used to determine the status of a Payment.  \*This field is only applicable to responses with  PaymentTypeID ( 3,4,5,6,7,D,O ) |
| CVVResult | string | 1 | The CVV result for matching the verfiication value.  This field is for infromation purposes and is not to be used to determine the status of a Payment.  \*This field is only applicable to responses with  PaymentTypeID ( 3,4,5,6,7,D,O ) |
| VANReference | string | 10 | The Gateway transaction reference to be used later |
| TransactionID | string |  | in captures, voids, and credits. |
| Last4 | string |  | The unique transaction indentifier assigned by the |
| PaymentDescription | string |  | The description of the account data.  Credit Card payments will be the First 6 digits +  masked digits + Last 4 digits  ACH payments will be the routning number + space  + masked account digsts + Last 4 account digits |
| Amount | decimal |  | Amount of the payment |
| PaymentTypeID | string | 1 | The payment type idnetifier  3=American Express  4=Visa  5=MasterCard  6=Discover  7=JCB  D=Debit Card O=Other C=ACH  \*Additional values could be added when future payment types become available to the system |
| EntryMode | String | 1 | The method of entry:  K = Manually Keyed  H = Track 1  D = Track 2  M = MICR Read  S = Swipe  I = Insert  C = Contactless  F = Fallback swipe |
| NetworkID | String | 1 | P = Paymentech  T = TSYS  E = Elavon  F = First Data  S = SPS ACH  \*Additional values could be added when future networks become available to the system. |
| CardExpirationDate | string | 4 | The date the account data will expire, format MMYY. |
| CurrencyCode | string | 3 | Represents the transaction currency. USD = 840. |
| SignatureImageData | string |  | Base-64 encoded image data, representing the transaction signature. |
| SignatureFormat | string |  | Signature image format; eg, “PNG” |

EmvResponseType element

<EmvResponseType>

<EmvTags></EmvTags>

</EmvResponseType>

The EmvResponseType element contains the response elements related to an EMV transaction.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| EmvTags | EmvTagType | Array | Yes | Contains an array of key/value EmvTagType elements. Please see the EMV-specific documentation for more information. |

MonerisReceiptType element

<MonerisReceipt>

<Account></Account>

<Amount></Amount>

<AppLabel></AppLabel>

<AppPreferredName></AppPreferredName>

<ARQC></ARQC>

<AuthCode></AuthCode>

<CardNumber></CardNumber>

<CardType></CardType>

<CvmIndicator></CvmIndicator>

<EmvAID></EmvAID>

<InvoiceNumber></InvoiceNumber>

<IsoCode></IsoCode>

<MerchantAddress></MerchantAddress>

<MerchantCity></MerchantCity>

<MerchantName></MerchantName>

<MerchantState></MerchantState>

<MerchantZip></MerchantZip>

<Message></Message>

<PanEntry></PanEntry>

<RefNum></RefNum>

<ResponseCode></ResponseCode>

<TipAmount></TipAmount>

<TransDate></TransDate>

<TransTime></TransTime>

<TransType></TransType>

<TSI></TSI>

<TvrARQC></TvrARQC>

<TvrTCACC />

<Type></Type>

</MonerisReceipt>

The MonerisReceiptType element contains the response elements from the Moneris receipt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Account | String |  | No | Account type such as VISA, MasterCard etc. This is the full description of the CardType field. |
| Amount | String | 10 | No | Receipt Total amount. The amount represents the amount that the cardholder was charged/refunded. |
| AppLabel | String | 16 | No | Mnemonic associated with the AID. The Application Label contains only characters in the common character set that all EMV-capable terminals are required to support. |
| AppPreferredName | String | 16 | No | Preferred mnemonic associated with the AID. The Application Preferred Name is the name of the card application in the cardholder's local language. |
| ARQC | String | 16 | No | The Authorization Request Cryptogram (ARQC) is used for an Online Card Authentication process. This field is generated by the card for transactions that require online authorization. The issuer validates the ARQC to ensure the authenticity of the card. |
| AuthCode | String | 8 | No | The authorization code is returned by the issuer as part of the transaction response. |
| CardNumber | String | 19 | No | Masked card number. Only the last 4 digits are unmasked. |
| CardType | String | 2 | No | CardType indicator defines the card type used for a transaction.  M = MasterCard  V = Visa  AX = American Express  NO = Novus/Discover in (Canada only)  DS= Discover (US only)  C = JCB (US only)  C1 = JCB (Canada only)  SE = Sears (Canada only)  P = Pin Debit (US only)  D = Debit (Canada only) |
| CvmIndicator | String | 2 | No | The Cardholder Verification Method (CVM) field. EMV transactions with successful PIN entry will include the phrase “VERIFIED BY PIN) on the merchant copy. EMV transactions with unsuccessful PIN entry will include a signature line on the merchant copy. A signature line will also print if the initial CVM is Signature (returns “S”). Both “VERIFIED BY PIN” and a signature line will be included if the CVM is PIN & Signature (returns “B”). |
| EmvAID | String | 32 | No | The Application Identifier (Aid) field is a data label that identifies an application on the card or terminal. The Aid is used to determine which applications are supported as both the card and the terminal must support the same Aid to initiate a transaction. |
| InvoiceNumber | String |  | No | Invoice Number |
| IsoCode | String | 2 | No | The ISO code is a bank issued response code. The ISO code and the response code are used to determine the receipt message. |
| MerchantAddress | String |  | No | Merchant Address |
| MerchantCity | String |  | No | Merchant City |
| MerchantName | String |  | No | Merchant Name |
| MerchantState | String | 2 | No | Merchant State |
| MerchantZip | String |  | No |  |
| Message | String | 100 | No | The response message indicating in details whether the transaction was Approved or Declined |
| PanEntry | String | 1 | No | The PAN entry mode characters (S, M, C, F, G, T, H or Q):  S (swiped)  M (manual entry)  C (chip transaction)  F (fallback to swipe)  G (fallback to manual entry)  T (contactless tab nin EMV)  H (EMV contactless)  Q (chip card malfunction) |
| RefNum | String | 18 | No | Bank transaction reference number. The following example illustrates the breakdown of this field where "210123450010690030” is the reference number returned in the message:  Example: 210123450010690030  •21012345: Terminal ID  •001: Shift number  •069: Batch number  •003: Transaction number within the batch. |
| ResponseCode | String | 3 | No | The response code is used to determine whether the transaction was approved or declined. The receipt message is determined by the Response Code and the ISO code.  0 – 49 (inclusive) – Approved  50 – 999 (inclusive) - Declined  Null – Incomplete  For a full list of response codes and the associated message please refer to the [Financial Response Codes](https://developer.moneris.com/Documentation/NA/Card%20Present%20Solutions/Mobile/~/link.aspx?_id=E63D673AD4884EFD945506429503645E&_z=z). |
| TipAmount | String |  | No | Tip Amount |
| TransDate | String | 10 | No | Date of the transaction |
| TransTime | String | 8 | No | Time of the transaction. |
| TransType | String |  | No | Transaction Type defines what type of transaction was performed:  00 – Sale  01 – Authorization (Credit Cards only)  02 – Completion (Credit Cards only)  04 – Refund / Independent Refund  11 – Void |
| TSI | String | 4 | No | Indicates the functions performed in a transaction. |
| TvrARQC | String | 10 | No | The values of the flags in the EMV Terminal Verification Results (TVR) indicate whether unsafe conditions arose during the transaction. |
| TvrTCACC | String | 10 | No | The final Terminal Verification Results (TVR). |
| Type | String |  | No | Full description of the performed transaction type. |

BatchResponseType element

<BatchResponseType>

<Response></Response>

<BatchNumber></BatchNumber>

<BatchReference></BatchReference>

<Net></Net>

<Count></Count>

<BatchPayment></BatchPayment>

</BatchResponseType>

The BatchResponseType element contains the the response elements related to processing a batch.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Response | ResponseType |  | Yes | Contains the response elements related to the Gateway request |
| BatchNumber | String | 6 | Yes | The batch sequence number |
| BatchReference | String | 10 | Yes | The unique Gateway batch identifier |
| Net | decimal |  | Yes | The net transaction amount of the batch, a negative amount is possible when processing credits/refunds |
| Count | integer |  | Yes | The total transaction count of the batch |
| BatchPayment | BatchPaymentType |  | Yes | CREDITCARD  PURCHASECARD |

TransactionSettlementStatusType element

<TransactionSettlementStatusType>

<TransactionType></TransactionType>

<SettlementType></SettlementType>

<SettlementDate></SettlementDate>

<BatchReference></BatchReference>

</TransactionSettlementStatusType>

The TransactionSettlementStatusType element contains the settlement status elements related to a transaction.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| TransactionType | integer |  | Yes | The transaction type of the transaction processed |
| SettlementType | integer |  | Yes | The settlement type of the transacton processed  0 = Error / Declined  1= Batch  2 = Settled  3 = Expired |
| SettlementDate | string |  | No | The settlement date, if settled, for the transaction processed; MM/DD/YYYY HH:MM:SS |
| BatchReference | string | 10 | No | The batch reference, if settled, for the transaction processed |

TransactionStatusQueryResponseType element

<TransactionStatusQueryResponseType>

<Response></Response>

<VaultResponse></VaultResponse>

<RecurringResponse></RecurringResponse>

<TransactionResponse></TransactionResponse>

<TransactionSettlementStatus></TransactionSettlementStatus>

<Customer></Customer>

</TransactionStatusQueryResponseType>

The TransactionStatusQueryResponseType element contains the response elements related to processing a transaction.

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Required | Comments |
| Response | ResponseType | Yes | Contains the response elements related to the Gateway request |
| VaultResponse | VaultResponseType | No | Contains the response elements related to the Vault operation |
| RecurringResponse | RecurringResponseType | No | Contains the response elements related to the transaction enlistment into the recurring system |
| TransactionResponse | TransactionResponseType | Yes | Contains the response elements realted to processing a transaction on the Gateway |
| TransactionSettlementStatus | TransactionSettlementType | Yes | Contains the transaction settlement status |
| Customer | PersonType | Yes | Contains the billing/customer elements used for the payments |

RecurringStatusQueryResponseType element

<RecurringStatusQueryResponseType>

<Responses></Responses>

<TransactionResponses></TransactionResponses>

<TransactionSettlementStatuses></TransactionSettlementStatuses>

<Customers></Customers>

</RecurringStatusQueryResponseType>

The RecurringStatusQueryResponseType element contains the response elements related to a recurring transaction processed by the Sage system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Responses | ResponseType | Array | Yes | Contains the response elements related to the Gateway request |
| TransactonResponses | TransactionResponseType | Array | Yes | Contains the response elements related to the Vault operation |
| TransactionSettlementStatuses | TransactionSettlementType | Array | Yes | Contains the response elements related to the transaction enlistment into the recurring system |
| Customers | PersonType | Array | Yes | Contains the response elements realted to processing a transaction on the Gateway |

VaultAccountResponseType element

<VaultAccountResponseType>

<Response></Response>

<VaultAccount></VaultAccount>

<Merchant></Merchant>

</VaultAccountResponseType>

The RecurringStatusQueryResponseType element contains the response elements related to a recurring transaction processed by the Sage system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Response | ResponseType |  | Yes | Contains the response elements related to the account creation request |
| VaultAccount | VaultAccountType |  | Yes | Contains the account elements provided in the Request |
| Merchant | TransactionSettlementType | Array | Yes | Contains the newly created MerchantID and |

AccountQueryResponseType Element

<AccountQueryResponseType>

<Response></Response>

<Merchant></Merchant>

<Services></Services>

<Active></Active>

</AccountQueryResponseType>

The AccountQueryResponseType element contains the response elements related to a merchant account status and service list inquiry.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Data Type | Length | Required | Comments |
| Response | ResponseType |  | Yes | Contains the response elements related to the account inquiry request |
| Merchant | MerchantType |  | Yes | Contains the account elements provided in the request |
| Services | String | Array | Yes | Contains the an array of service descriptions  available to a merchant account.  VAULT = “Vault storage service”  CREDITCARD = “Credit Card payment service” |
| Active | Boolean |  | Yes | The status of the merchant account |

Field validation

XML reserved characters must be observed and encoded appropriately.

Alpha numeric fields\*

The following is a list of accepted characters for Alpha Numeric fields:

White Space a-z

A-Z

0-9

-.,#&()/!'éèêëòóôõöàáâãäåìíîïùúûüýÿ

Numeric fields\*

The following is a list of accepted characters for Numeric fields:

0-9

,.-

\*The use of two or more dashes -- back to back is not permitted in any of the fields.

Gateway error codes

|  |  |  |
| --- | --- | --- |
| Code | Message | Description |
| 000000 | INTERNAL SERVER ERROR | Server Error |
| 900000 | INVALID T\_ORDERNUM | Order number value is in an invalid format |
| 900001 | INVALID C\_NAME | Name value is in an invalid format or was left blank |
| 900002 | INVALID C\_ADDRESS | Address value is in an invalid format or was left blank |
| 900003 | INVALID C\_CITY | City value is in an invalid format or was left blank |
| 900004 | INVALID C\_STATE | State value is in an invalid format or was left blank |
| 900005 | INVALID C\_ZIP | Zip code value is in an invalid format or was left blank |
| 900006 | INVALID C\_COUNTRY | Country value is in an invalid format or was left blank |
| 900007 | INVALID C\_TELEPHONE | Telephone value is in an invalid format or was left blank |
| 900008 | INVALID C\_FAX | Fax value is in an invalid format or was left blank |
| 900009 | INVALID C\_EMAIL | Email value is in an invalid format or was left blank |
| 900010 | INVALID C\_SHIP\_NAME | Shipping address name value is in an invalid format |
| 900011 | INVALID\_C\_SHIP\_ADDRESS | Shipping Address value is in an invalid format |
| 900012 | INVALID\_C\_SHIP\_CITY | Shipping city value is in an invalid format |
| 900013 | INVALID\_C\_SHIP\_STATE | Shipping state value is in an invalid format |
| 900014 | INVALID\_C\_SHIP\_ZIP | Shipping zip code value is in an invalid format |
| 900015 | INVALID\_C\_SHIP\_COUNTRY | Shipping country value is in an invalid format |
| 900016 | INVALID\_C\_CARDNUMBER | Credit card number value is in an invalid format |
| 900017 | INVALID\_C\_EXP | Expiration date value is in an invalid format |
| 900018 | INVALID\_C\_CVV | CVV (card verification value) value is in an invalid format or was left blank (if set to required) |
| 900019 | INVALID\_T\_AMT | Grand Total must equal > $0.00. Please check subtotal, shipping and tax values. |
| 900020 | INVALID\_T\_CODE | Transaction Code value is in an invalid format or was left blank |
| 900021 | INVALID\_T\_AUTH | Authorization code is in an invalid format or was left blank (required for Force transactions) |
| 900022 | INVALID\_T\_REFERENCE | Reference value is in an invalid format or was left blank (Required for Force or Void by Reference) |
| 900023 | INVALID\_T\_TRACKDATA | Track Data value is in an invalid format or was left blank (required for debit and retail transactions) |
| 900024 | INVALID\_T\_TRACKING\_NUMBER | Tracking number value is in an invalid format |
| 900025 | INVALID\_T\_CUSTOMER\_NUMBER | Customer number value is in an invalid format(used only for PCLIII transactions) |
| 900026 | INVALID\_T\_SHIPPING\_COMPANY | Shipping company value is in an invalid format |
| 900027 | INVALID\_T\_RECURRING | Recurring value is in an invalid format (must be = 0 or 1) |
| 900028 | INVALID\_T\_RECURRING\_TYPE | Recurring value is in an invalid format |
| 900029 | INVALID\_T\_RECURRING\_INTERVAL | Recurring interval value is in an invalid format (must be numeric) |
| 900030 | INVALID\_T\_RECURRING\_INDEFINITE | Recurring indefinite value is in an invalid format or was left blank |
| 900031 | INVALID\_T\_RECURRING\_TIMES\_TO\_PROCESS | Recurring times to process value is in an invalid format (must be numeric) |
| 900032 | INVALID\_T\_RECURRING\_NON\_BUSINESS\_DAYS | Recurring non business days value is in an invalid format |
| 900033 | INVALID\_T\_RECURRING\_GROUP | Recurring Group was left blank or group not found |
| 900034 | INVALID\_T\_RECURRING\_START\_DATE | Recurring start date value is in an invalid format or was left blank |
| 900035 | INVALID\_T\_PIN | Pin number entered is incorrect (required for Pin- debit transactions) |
| 901000 |  | General data validation error, the message will contain additional information |
| 910000 | SERVICE NOT ALLOWED | The transaction you are trying to submit is not allowed. |
| 910001 | VISA NOT ALLOWED | Visa card type transactions are not allowed. |
| 910002 | MASTERCARD NOT ALLOWED | MasterCard card type transactions are not allowed. |
| 910003 | AMEX NOT ALLOWED | American Express card type transactions are not allowed. |
| 910004 | DISCOVER NOT ALLOWED | Discover card type transactions are not allowed. |
| 910005 | CARD TYPE NOT ALLOWED | Card type transactions are not allowed. |
| 911911 | SECURITY VIOLATION | M\_id or M\_key incorrect |
| 920000 | ITEM NOT FOUND | Item not found |
| 920001 | CERDIT VOL EXCEEDED | No corresponding sale found within last 6 months, credit couldn’t be issued. |