SMARTVISTA EXCHANGE PROTOCOL WITH OMNICHANNEL (SVXP OMNI ISS)

API developer reference

April 2018

Contents

[1 PREFACE 3](#_Toc520126778)

[1.1 Revision history 3](#_Toc520126779)

[1.2 Document purpose 3](#_Toc520126780)

[2 SMARTVISTA INTEGRATION SERVICES OVERVIEW 3](#_Toc520126781)

[2.1 General concepts 3](#_Toc520126782)

[2.2 Data types, Occurrence, Dictionaries 3](#_Toc520126783)

[3 CARD FILE STRUCTURE 4](#_Toc520126784)

[3.1 Overview 4](#_Toc520126785)

[3.2 References 4](#_Toc520126786)

[3.3 List of elements 5](#_Toc520126787)

[4 PRODUCT FILE STRUCTURE 6](#_Toc520126788)

[4.1 Overview 6](#_Toc520126789)

[4.2 References 7](#_Toc520126790)

[4.3 List of elements 7](#_Toc520126791)

1. PREFACE
   1. Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Author | | Details |
| 1.0 | 16.04.2018 | | Kolodkina Y. | Initial version |
| 2.0 | 23.07.2018 | | Kolodkina Y. | Added Product file structure |

* 1. Document purpose

SVXP OMNI is a reference manual for developers who are implementing API of the SmartVista solution. This document is written for internal use of BPC. The document describes the content and the structure of the API.

It is supposed document users to be familiar with financial transactions, communications and XML data format.

1. SMARTVISTA INTEGRATION SERVICES OVERVIEW
   1. General concepts

SmartVista exchange protocol with OmniChannel (SVXP OMNI hereafter) provides a description of the file formats of information upload into OmniChannel (OMNI) from SmartVista. File format xml. For each format will be described in this document, XML Schema Definition language (XSD) and provided examples.

* 1. Data types, Occurrence, Dictionaries

For SVXP OMNI methods the standard XML data types are used. Those are fully described in the following document**XML Schema Part 2: Datatypes Second Edition and** t can be found here: [***http://www.w3.org/TR/xmlschema-2/***](http://www.w3.org/TR/xmlschema-2/)

Within the current document all the SVXP OMNI messages are described in the table structure below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag name | Data type | Length | Occurrence (min-max) | Description |
| Card | | | | |
| **card\_id** | string | 200 | 1-1 | Internal card instance identifier(UID) |
| inst\_id | int | 4 | 1-1 | An institution ID |
| card\_number | string | 24 | 1-1 | A card number(PAN). It must be unique in the scope of an institution. It can be masked. |
| card\_mask | string | 24 | 0-1 | Masked card number |
| card\_type | int | 4 | 0-1 | Card type: Visa Gold, Visa Classic etc. Dictionary value |
| category | string | 8 | 0-1 | Category of card. Value from dictionary CRCG |
| Lang | String | 7 | 0-1 | Language. Please refer to LANG dictionary |

**Data Type:** SVXP OMNI tags can be of Primitive XML Data Types (string, long, boolean, etc ) or Complex Data Types (Aggregates).

**Occurrence**: This field defines if the field is mandatory or optional (first number) as well as maximum number occurrences of this tag in the message (last digit)

e.g. 1-1 = minOccurs="1" maxOccurs="1"

Documentation is provided along with Examples of the Request messages for all of the methods described below.

1. CARD FILE STRUCTURE
   1. Overview

The file contains information on cards. Including card data and card services data. The information about cards is synchronized between the SmartVista and OmniChannel.

The direction of the file may be only OUTGOING.

Tag CARDS is root tag and it include itself one or more tags CARD.

* 1. References

Format of card file described by XSD file.

SVXP OMNI XSD: svxp\_omni\_iss\_card.xsd

Example of xml document:

svxp\_omni\_iss\_card.xml – cards information.

* 1. List of elements

| Tag name | Data type | Length | Is mandatory | Description |
| --- | --- | --- | --- | --- |
| cards | | | | It is root tag and it include itself one or more tags card |
| file\_id | long | 16 | 0-1 | Unique identifier of outgoing file |
| file\_type | string | 8 | 1-1 | Type of outgoing file. Describe the purpose of data in file. Dictionary FLTP. Value FLTPCSIF. |
| inst\_id | int | 4 | 0-1 | An institution ID |
| card | complex |  | 1-\* | Card data. |
| card | | | | |
| card\_id | string | 200 | 1-1 | Internal card instance identifier(UID) |
| card\_number | string | 24 | 1-1 | A card number(PAN). It must be unique in the scope of an institution. It can be masked. |
| card\_type | int | 4 | 0-1 | Card type: Visa Gold, Visa Classic, MasterCard Gold etc. |
| customer | complex |  | 1-1 | Customer-owner of card. Should contain an attribute customer\_id |
| contract | complex |  | 1-1 | Contract by which card was registered. Should contain an attribute contract\_id |
| available\_services | complex |  | 1-\* | Available services of card |
| attached\_services | complex |  | 0-\* | Attached services of card |
| customer | | | | |
| customer\_id | Long | 12 | 1-1 | Unique identifier of customer in SVBO. It can be used for search and update/insert data it OMNI |
| customer\_number | String | 200 | 1-1 | An external ID of a customer. It must be unique in the scope of an institution |
| contract | | | | |
| contract\_id | Long | 12 | 1-1 | Unique identifier of contract in SVBO. It can be used for search and update/insert data it OMNI |
| contract\_number | String | 200 | 1-1 | A contract number. It must be unique in the scope of an institution |
| available\_services | | | | |
| service\_type | string | 8 | 1-1 | Service type |
| service\_type\_name | string | 200 | 1-1 | Service name |
| service\_external\_code | string | 200 | 0-1 | Service external code |
| service\_number | string | 200 | 0-1 | Service number |
| attached\_services | | | | |
| service\_type | string | 8 | 1-1 | Service type |
| service\_type\_name | string | 200 | 1-1 | Service name |
| service\_external\_code | string | 200 | 0-1 | Service external code |
| service\_number | string | 200 | 0-1 | Service number |

1. PRODUCT FILE STRUCTURE
   1. Overview

This file is used for export product’s. In this format products and their services, cards and accounts could be described.

The direction of the file may be only OUTGOING.

Tag PRODUCTS is root tag and it include itself one or more tags PRODUCT.

* 1. References

SVXP OMNI Web Services schema and its underlying components are required for SmartVista. The following are the locations of the WSDL file.

SVXP OMNI WSDL: svxp\_omni\_product.wsdl

Format of product file described by XSD file: svxp\_omni\_product.xsd

Example of xml document: svxp\_omni\_product\_example.xml

* 1. List of elements

| Tag name | Data type | Length | Occurs | Description |
| --- | --- | --- | --- | --- |
| products | | | | |
| file\_type | string | 8 | 1-1 | Type of outgoing file. Dictionary FLTP. Value FLTPPROD. |
| inst\_id | int | 4 | 1-1 | Identifier of institution which owns products in file. |
| file\_id | long | 16 | 1-1 | Unique identifier of outgoing file. |
| product | complex |  | 1-\* | Encapsulate all information about one product (with subproducts). |
| product | | | | |
| product\_type | string | 8 | 1-1 | Indicates the type of product. Values are taken from the dictionary PRDT. |
| contract\_type | string | 8 | 1-1 | Indicates the type of contract which can be serviced by the product. Values are taken from the dictionary CNTP. |
| customer\_types | complex |  | 1-\* | Indicates the types of customer which can be serviced by the product. List of allowable value. Values are taken from the dictionary ENTT. |
| product\_number | string | 200 | 1-1 | Unique product number. |
| product\_name | product\_name |  | 0-1 | Product name that may be presented in different languages at once. This tag must contain the LANGUAGE attribute to display them in the proper language. |
| product\_status | string | 8 | 1-1 | Activity status of product. Values are taken from the dictionary PRDS. |
| product\_service | product\_service |  | 0-\* | List of services which are configured on product. |
| product\_accounts | complex |  | 0-\* | List of account types on product. |
| product\_cards | complex |  | 0-\* | List of cards types on product. |
| product | product |  | 0-\* | Child products of this product. |
| customer\_types | | | | |
| customer\_type | string | 8 | 1-1 | Customer type. Values are taken from the dictionary ENTT. |
| product\_name | | | | |
| name | string | 200 | 1-1 | Short name of product. |
| description | string | 2000 | 0-1 | Description of product. |
| product\_service | | | | |
| service\_number | string | 200 | 1-1 | Unique service number connected or must be connected to product. |
| service\_type\_id | int | 8 | 1-1 | Reference to service type. |
| service\_status | string | 8 | 1-1 | Activity status of service. Values are taken from the dictionary SRVS. |
| min\_count | int | 4 | 1-1 | Minimum of services opened for one contract. |
| max\_count | int | 4 | 1-1 | Maximum of services opened for one contract. |
| service\_name | service\_name |  | 0-999 | Service name that may be presented in different languages at once. This tag must contain the LANGUAGE attribute to display them in the proper language. |
| service\_name | | | | |
| name | string | 200 | 1-1 | Short name of service. |
| description | string | 2000 | 0-1 | Description of service. |
| product\_accounts | | | | |
| account\_type | string | 8 | 1-1 | Indicates the type of account. Values are taken from the dictionary ACTP. |
| currency | string | 3 | 1-1 | Currency code of account. |
| service\_number | string | 200 | 1-1 | Unique identifier of initial service of account which is connected or must be connected to product. |
| product\_cards | | | | |
| card\_type\_id | int | 4 | 1-1 | Type of card. |
| service\_number | string | 200 | 1-1 | Unique identifier of initial service of card which is connected or must be connected to product. |