**Asurion Subscriber Billing System Finance Services API Specification**

**Versio****n: 2.0**

**Revision and Sign-off Sheet**

Change Record

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
| 11/28/2011 | Jim Emerson | 0.1 | Draft version |
|  |  |  |  |
| 12/13/2011 | Jim Emerson | 1.0 | All corrections and changes included. |
| 2/11/2012 | Jim Emerson | 2.0 |  |
|  |  |  |  |

Document Properties

| Item | Details |
| --- | --- |
| Document Title | Asurion Subscriber Billing System Finance Services API Specification |
| Author | Jim Emerson |
| Creation Date | 11/28/2011 |
| Last Updated | 2/11/2012 |

**Table of Contents**

[Introduction 4](#_Toc41637596)

[Finance Services 4](#_Toc41637597)

[Finance Service Gateway 5](#_Toc41637598)

[Asurion Proxy Server Process Flow 7](#_Toc41637599)

[Virtual IP (VIP) Process Flow 7](#_Toc41637600)

[GetContracts Process Flow 7](#_Toc41637601)

[FinanceServiceAgent Process Flow 7](#_Toc41637602)

[CreateContractGateway Process Flow 9](#_Toc41637603)

[ProcessNonRefPaymentGateway Process Flow 11](#_Toc41637604)

[CalculateTaxesGateway Process Flow 12](#_Toc41637605)

[API Specification 13](#_Toc41637606)

[Finance Service API Specification 13](#_Toc41637607)

[Finance Service WSDL Definition 13](#_Toc41637608)

[Finance Service Gateway APIs 14](#_Toc41637609)

[Finance Service Gateway Databases 16](#_Toc41637610)

[Asurion Finance Database 16](#_Toc41637611)

[CRM Database 17](#_Toc41637612)

[Appendix 18](#_Toc41637613)

[A.1 Asurion Finance Service Gateway API Description 18](#_Toc41637614)

[A.1.1 CreateContract Service API 18](#_Toc41637615)

[A.1.2 ProcessNonRefPayment Service API 20](#_Toc41637616)

[A.1.3 CalculateTaxes Service API 22](#_Toc41637617)

[A.3 Example: CreateContact Request to Create a Billing Account 25](#_Toc41637618)

[Index 27](#_Toc41637619)

**Figures**

[Figure 1. Asurion Finance Service Gateway 6](#_Toc41637620)

[Figure 2. GetContract Process Flow 7](#_Toc41637621)

[Figure 3. Asurion Finance Service Agent Process Flow 8](#_Toc41637622)

[Figure 4. CreateContract Process Flow 9](#_Toc41637623)

[Figure 5. ProcessNonRefPayment Gateway Process Flow 11](#_Toc41637624)

[Figure 6. CalculateTaxes Gateway Process Flow 12](#_Toc41637625)

# Introduction

This document describes the set of APIs that are central to the Asurion Subscriber Billing System. The system supports both automatic and UI controlled service operations. Automatic “CRM” processes flow after a customer enrollment through Asurion Finance Service Gateway actions. Authorized users control manual “NET” processes via Tibco BusinessWorks (BW) screens. These services are accessed through the CreateContract, ProcessNonRefPayment, and CalculateTaxes Gateways.

## Finance Services

The Asurion Subscriber Billing System incorporates the following specialized Finance Services:

* **CalculateTaxes**—Asurion uses this service to process requests for tax calculations and return responses. See “Figure 6. CalculateTaxes Gateway Process Flow” in this document.
* **CalculateTaxesGateway**—Gateway for processing tax calculation request and responses. See “
* CalculateTaxesGateway Process Flow” and “Figure 6. CalculateTaxes Gateway Process Flow” in this document.
* **CreateContract Gateway**—Gateway for processing requests to Create Contracts. See “CreateContractGateway Process Flow” and “Figure 4. CreateContract Process Flow” in this document.
* **FinanceGateway**—Gateway for processing financial transaction requests and responses. See “Figure 1. Asurion Finance Service Gateway” in this document.
* **FinanceServiceAgent**—Asurion uses the FinanceServiceAgent to manage all requests and responses passed through the Finance Service Gateway. See “Also See
* Background information on the Aria standards used by Asurion to link Subscriber Billing System features to Aria, see the “Aria Web Services API QuickStart Guide” at https://oneteam/Teams/OrganizationalTeams/TechnologyProducts/RelationshipProductMgmt/ProductMgmtBackOffice/HRFinanceLegal/SubscriberBilling/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2FTeams%2FOrganizationalTeams%2FTechnologyProducts%2FRelationshipProductMgmt%2FProductMgmtBackOffice%2FHRFinanceLegal%2FSubscriberBilling%2FShared%20Documents%2FAsurion%20Subscriber%20Billing%20System%20Specs&FolderCTID=0x0120005617996FC3AE2747809B62081EE5F1B4&View={6DA69BD2-25DE-48A5-A0B8-039239D5C7F8}.
* FinanceServiceAgent Process Flow” in this document.
* **GetContract**—Aria uses the GetContract service to get Asurion information on contracts. See “Figure 2. GetContract Process Flow” in this document.
* **ProcessNonRefPayment**— Asurion uses the ProcessNonRefPayment service to process non-refndable payments. See “Figure 5. ProcessNonRefPayment Gateway Process Flow” in this document.
* **ProcessNonRefPaymentGateway**—Gateway for processing non-refundable payments. See ““Figure 5. ProcessNonRefPayment Gateway Process Flow” in this document.
* **ProcessNonRefPremium Payment**—Asurion uses this service to process non-refundable premium payments. See “ Figure 4. CreateContract Process Flow” in this document.

**Note**: These are the basic Asurion Subscriber Billing System. Some customers may incorporate other Asurion services (e.g., ProcessPremium) into their implementations.

**Also See**

See the “Asurion Subscriber Billing System Product Specification” for a list of all possible services.

# Finance Service Gateway

Asurion Finance Services includes a multi-functional Finance Services Gateway (also known as the “Gateway”). The Gateway incorporates a Finance Service Agent with methods that interact with three specialized API gateways:

* **CreateContractGateway**—Passes requests via the CreateContract Service to handle requests for contract creation. CreateContract is a TIBCO step in the Asurion enrollment process.
* **ProcessNonRefPaymentGateway**—Passes requests via the ProcessNonRefPayment Service to handle requests to process non-refunded payments and return responses.
* **CalculateTaxesGateway**—Passes requests via the CalculateTaxesService to handle requests for tax calculations and return responses.

These process flows are represented below.

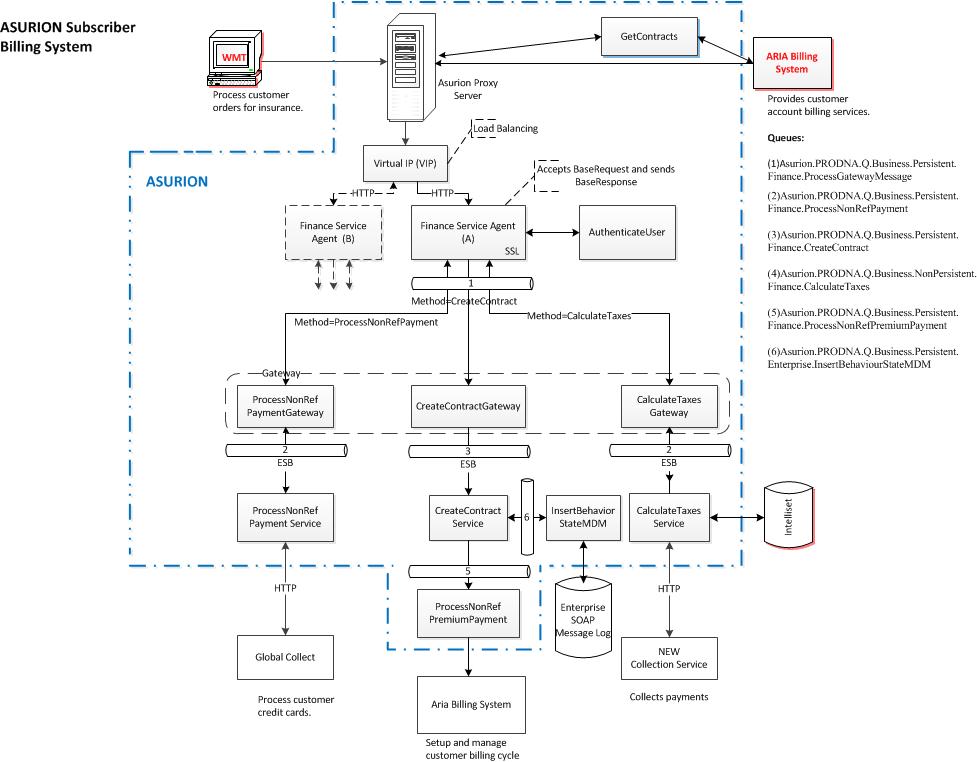


Figure 1. Asurion Finance Service Gateway

The components of these flows are described in the following sections.

### Asurion Proxy Server Process Flow

Within Asurion, the Proxy Server forwards requests from Wal-Mart to create CRM contracts for customer insurance. The Proxy Server also requests current customer contract information from the ARIA Billing Service via GetContracts. Handling create contract request, the Proxy Server uses the Virtual IP to pass requests to the appropriate Finance Service Agent for processing.

### Virtual IP (VIP) Process Flow

Providing load balancing, the Asurion Virtual IP (VIP) passes each request message from the Proxy Server to a Finance Service Agent. The VIP determines which Finance Service Agent will receive a message depending upon standard criteria (operational status, smallest current queue, and so forth).

### GetContracts Process Flow

The Asurion Proxy Server requests information (via HTTP) on current contracts through GetContracts. Aria responds with the latest customer contract information. Aria also sends requests to process non-refundable credit card payments to Asurion via GetContracts. This process flow is represented in the next figure.

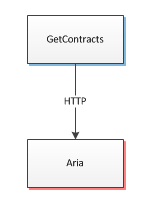


Figure 2. GetContract Process Flow

**Also See**

Background information on the Aria standards used by Asurion to link Subscriber Billing System features to Aria, see the “Aria Web Services API QuickStart Guide” at

**<Link Removed for Sample>**

### FinanceServiceAgent Process Flow

The Asurion Finance Service Agent (FSA) manages all requests and responses passed through the Finance Service Gateway. FSA accepts BaseRequest request messages from the VIP to returns BaseResponse response messages back to the server. Each BaseRequest includes request headers, URL arguments, and the submitted files. A BaseResponse includes response headers, current status, and response data as a string.

Upon accepting requests, FSA first authenticates the request user before further processing the request. FSA sends an AUTH message to the AuthenticateUser Service for user verification. This service verifies the user ID based on information stored in an Active Directory and returns the results to FSA.

When user authentication is returned, FSA sends each request to a central queue (Asurion.PRODNA.Q.Business.Persistent.Finance.ProcessGatewayMessage) which calls the API methods CreateContract, ProcessNonRefPayment, or CalculateTaxes to pass the request to the appropriate service. In the opposite direction, FSA accepts return responses from the ProcessNonRefPaymentGateway (via the ProcessNonRefPayment method) and CalculateTaxesGateway (via the CalculateTaxes method). The FSA has a set of global variables that are available to Gateway APIs. The FSA process flow is represented in the next figure.

**Note**: The CreateContract process flow path is one-way (request only).

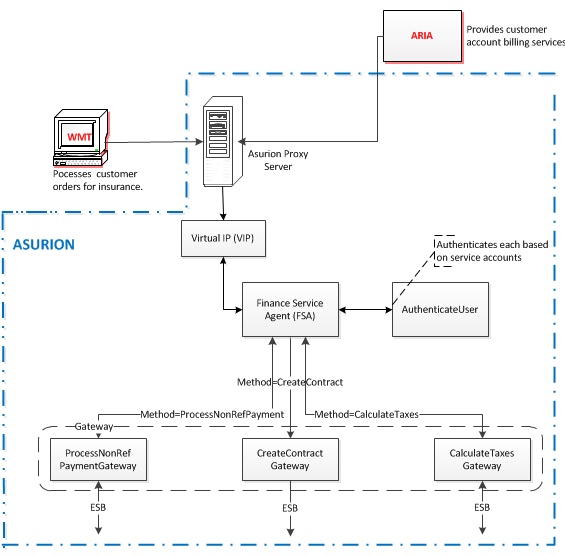


Figure 3. Asurion Finance Service Agent Process Flow

### CreateContractGateway Process Flow

Upon accepting a request (SOAP via HTTP) to create a CRM customer contract from the FSA via the API method CreateContract, the CreateContractGateway adds the request to a message queue (Asurion.PRODNA.Q.Business.Persistent.Finance.CreateContract). From the queue, the request passes via an Enterprise Service Bus (ESB) to the CreateContract Service for processing.

**Note**: ESB enables the gateway to handle messages that originated with different protocols (HTTP, FTP, SMTP, and so forth).

These process flows are represented in the following figure.

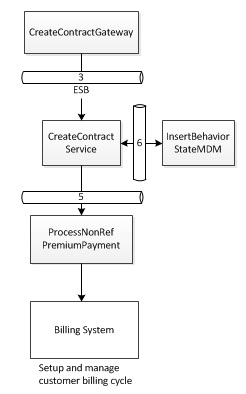


Figure 4. CreateContract Process Flow

#### CreateContract Service

Upon receiving a request from the message queue, the CreateContract Service first sends a request to the InsertBehaviorStateMDM (via the message queue Asurion.PRODNA.Q.Business.Persistent.Finance.Enterprise.InsertBehaviorStateMDM) to determine the current behavior state of the request to prevent processing duplication.

Upon receiving the message, InsertBehaviorStateMDM queries the Enterprise SOAP Message Log for log messages about the current values of “BehaviorInteractionIdentifier” and “CurrentBehaviorInteractionIdentifier” and returns the status information to InsertBehaviorStateMDM via the message queue.

If the status information indicates that the create contract request has not already advanced to the next process, CreateContract Service sends the create contract request via a message queue (Asurion.PRODNA.Q.Business.Persistent.Finance.ProcessNonRefPremiumPayment) to the ProcessNonRefPremiumPayment API method.

#### InsertBehaviorStateMDM

Upon receiving requests for the current status of a device from the CreateContractService, the InsertBehaviorStateMDM gets the information from the Enterprise SOAP Message Log (maintained by TIBCO) and passes it back to the service.

**Also See**

“CRM Database”.

#### ProcessNonRefPremiumPayment API

Upon receiving requests to process non-refundable credit card payments from the CreateContract Service, this API formulates the request and submits it to the outside Aria billing service.

**ARIA Billing System**

Outside Asurion, the Aria Billing System accepts requests to create customer contracts from the Asurion ProcessNonRefPremiumPayment API. In response, the Billing System establishes and manages customer billing cycles for Asurion insurance coverage. When the Asurion Proxy Server requests information (via HTTP) on current contracts through GetContracts, Aria responses with the latest customer contract information. Aria also sends requests to process non-refundable credit card payments to Asurion.

**Also See**

“Asurion Subscriber Billing System Product Specification” for descriptions of data types supported for these APIs.

### ProcessNonRefPaymentGateway Process Flow

Upon accepting a request (SOAP via HTTP) to process non-refundable credit card payments from the FSA via the API method ProcessNonRefPayment, the ProcessNonRefPayments Gateway adds the request to a message queue (Asurion.PRODNA.Q.Business.Persistent.Finance.ProcessNonRefPayment). From this queue, the request is passed via an ESB to the ProcessNonRefPayment Service.

These process flows are represented in the following figure.

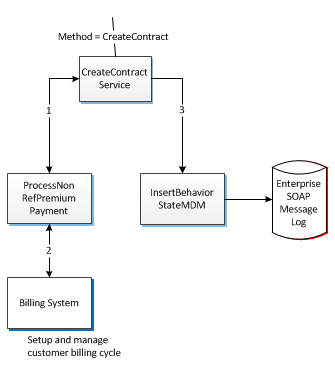


Figure 5. ProcessNonRefPayment Gateway Process Flow

#### ProcessNonRefPremiumPayment Service

Upon receiving a request to process a non-refundable credit card payment, this service sends it (via HTTP) to the outside Global Collect service. In the return direction, responses from the Global Collect are sent back (via HTTP) to the ProcessNonRefPayment Service which sends them up through the ESB to the queue, and finally, to the gateway.

#### Global Collect Service

Outside Asurion, the Global Collect service accepts requests from the ProcessNonRefPayment Service to process non-refundable credit card payments. Global Collects processes the requests and returns responses back (via HTTP) to the ProcessNonRefPayment Service.

### CalculateTaxesGateway Process Flow

Upon accepting a request (SOAP via HTTP) to calculate taxes on a customer bill from the FSA via the API method CalculateTaxes, the CalculateTaxesGateway adds it to a message queue (Asurion.PRODNA.Q.Business.Persistent.Finance.CalculateTaxes). From the queue, the request is passed via an ESB to the CalculateTaxes Service. These process flows are shown below.

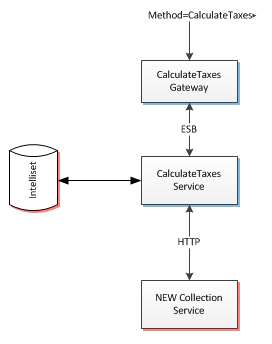


Figure 6. CalculateTaxes Gateway Process Flow

**Also See**

XML schema description in the section “A.1.3 CalculateTaxes Service API”.

#### CalculateTaxesService

Upon receiving a request to calculate taxes via the API method CalculateTaxes, the CalculateTaxes Service gets information about the rate of sales and service tax that should be charged for contracts purchased by customers and services (for example, maintenance) provided to customers. This data is included in a request sent (via HTTP) to the new collection service.

#### New Collection Service

This release of the Asurion Finance Services includes processing flows with a new collection service. Upon receiving a request to calculate taxes for a contract from the CalculateTaxes Service, this service processes the request and billing (including the taxes) for customer contracts that use the Vertex online Web-based billing service. (FSA does not connect to Vertex.)

# API Specification

This section introduces the APIs incorporated into the Asurion Subscriber Billing System, Finance Services. These component introductions focus on the Asurion Enterprise Canonical Model, FinanceService WSDL, and XML schema definitions of individual APIs.

## Finance Service API Specification

This section introduces the Asurion Finance Service API specification.

### Finance Service WSDL Definition

The Finance Service definition, included in the TIBCO XML WSDL developed with Asurion includes definitions of the service name, port name, binding, address location, port type, and included operations. A section of the WSDL is shown below.

<!--Created by TIBCO WSDL-->

…

<wsdl:service name="FinanceService">

<wsdl:port name="FinanceEndPoint" binding="tns:FinanceEndPointBinding">

<soap:address location="http://localhost:8080/services/Finance"/>

</wsdl:port>

</wsdl:service>

<wsdl:portType name="FinancePortType">

<wsdl:operation name="CalculateTaxes">

<wsdl:input message="tns:BaseRequest"/>

<wsdl:output message="tns:BaseResponse"/>

<wsdl:fault name="fault" message="tns:Errors"/>

</wsdl:operation>

<wsdl:operation name="ProcessNonRefPayment">

<wsdl:input message="tns:BaseRequest"/>

<wsdl:output message="tns:BaseResponse"/>

<wsdl:fault name="fault" message="tns:Errors"/>

</wsdl:operation>

<wsdl:operation name="CreateContracts">

<wsdl:input message="tns:BaseRequest"/>

<wsdl:output message="tns:BaseResponse"/>

<wsdl:fault name="fault" message="tns:Errors"/>

</wsdl:operation>

</wsdl:portType>

The following sections identify the data types included in the general Asurion Canonical Model and API’s CreateContract, CalculateTaxes, and ProcessNonRepayment.

**Also See**

“A.1 Asurion Finance Service Gateway API Description”.

### Finance Service Gateway APIs

The Finance Service Gateway incorporates three key APIs: CreateContract, ProcessNonRefPayments, and CalculateTaxes. The following sections introduce the APIs.

**Also See**

“Asurion Subscriber Billing System Product Specification”” for the data types supported for these APIs.

#### CreateContract API

The CreateContract service method is used to pass requests from the Finance Service Agent (FSA) to the CreateContract Gateway. This process involves:

* Creating contact details.
* Creating a contracted product list.
* Creating an enrollment.
* Calling the Asurion Finance Service to create a payment.

CreateContract Request

A component of the Asurion-Tibco enterprise message service, the CreateContractRequest API provides real time enrollment messaging between the WalMart point of sale and the Aria billing service. WalMart agents use the API to create new billing contracts in the Aria system.

**SOAP Request—**Click the following link to see a CreateContract SOAP request:

**<Link Removed for Sample>**

CreateContract Response

Another component of the Asurion-Tibco enterprise message service, the CreateContractResponse API returns the results to the FSA.

**SOAP Response—**Click the following link to see a CreateContract SOAP response:

**<Link Removed for Sample>**

**Also See**

Required and optional data types described in “A.2.2 Asurion Finance Services Data” in the “Asurion Subscriber Billing System Product Specification” and “A.1.1 CreateContract Service API” in this document.

#### ProcessNonRefPayment API

The ProcessNonRefPayment API service method is used to create and process ad hoc payment accounts for Echeck, CerditCard and Instore payments. The method takes requests from the Finance Service Agent (FSA) to the ProcessNonRefPayment Gateway and handles responses returning to the FSA.

ProcessNonRefPayment Request

Two types of ProcessNonRefPayment Request are possible: ProcessNonRefPaymentWithTender and ReProcessNonRefPaymentByTransactionId.

* ProcessNonRefPaymentWithTender—This request creates or processes an ad hoc payment using EcheckTender, InstoreTender, or CreditcardTender.
* ReProcessNonRefPaymentByTransactionId—This request reprocessed a previous payment using the previous payment transactionId.

**SOAP Request—**Click the following link to see a ProcessNonRefPayment SOAP Request.

**<Link Removed for Sample>**

ProcessNonRefPayment Response

The response uses ProcessNonRefPaymentResult to return a SOAP message. The Result contains processed transaction details and the valid payment status.

**SOAP Response—**Click the following link to see a ProcessNonRefPayment SOAP Response. **<Link Removed for Sample>**

**Also See**

Required and optional data types described in “A.2.2 Asurion Finance Services Data” in the “Asurion Subscriber Billing System Product Specification” and ”

A.1.2 ProcessNonRefPayment Service API” in this document.

#### CalculateTaxes API

The CalculateTaxes API method is used to pass requests from the Finance Service Agent (FSA) to the CalculateTaxes Gateway and responses back to the FSA.

CalculateTaxes Request

The SOAP message request to calculate taxes identifies the account by invoice number, invoice amount, invoice type, country code, and ship to address.

**SOAP Request—**Click the following link to see a CalculateTaxes SOAP Request:

**<Link Removed for Sample>**

CalculateTaxes Response

The SOAP message response to the request to calculate taxes on an account provides the following information: return code, invoice number, total tax amount, total tax rate, state tax amount, state tax rate, country tax amount, county tax rate, district tax amount, district tax rate, city tax amount, city tax rate, country name, geo code, PSR amount, QST amount, GST amount, and HST amount.

**SOAP Response**—Click the following link to see a CalculateTaxes SOAP Response:

**<Link Removed for Sample>**

**Also See**

Required and optional data types described in “A.2.2 Asurion Finance Services Data” in the “Asurion Subscriber Billing System Product Specification” and “A.1.3 CalculateTaxes Service API” in this document.

# Finance Service Gateway Databases

Finance Service Gateway API operations access several databases to store, update, and retrieve records primarily stored in two databases: Asurion Finance Database and the CRM Database. These databases are briefly described in the following sections.

**Note**: Gateway activities are enabled through databases for DAX, IntelliSet, reporting, enrollment, CRM, and other operations data stores (vSphere, SSIS, ETL, ESB, and so on). This section introduces the two primary databases used in Finance Service operations.

## Asurion Finance Database

This database stores Gateway-processed financial transaction records including message requests and responses. Gateway processes access the DB to display/update records. The database includes the following tables:

* **Transaction Table**—Stores data about a particular transaction, identified by a unique transaction ID. Other tables get information about the transaction from this table.
* **TransactionDetail Table**—Stores data about transactions. Other tables refer to fields in this table for transaction details.
* **ECheckDetail Table**—Stores data about e-check transactions.
* **CreditCardDetail Table**—Stores data about credit card transactions.
* **DirectDebitDetail Table**—Stores data about transactions processed through DDProcessing Gateway.
* **EDIFactDetail Table**—Stores data about transactions processed via EDIFact Gateway.
* **InStoreDetail Table**—Stores data about in-store transactions.
* **DeuCSDetail Table**—Stores data about transactions processed via DeuCS Gateway.
* **VerisignDetail Table**—Stores data that is used in transactions processed via PayFlowPro Gateway.
* **Refund Queue Table**—Stores data about refunds processed by the system.
* **BTADetail Table**—Stored data about the Bill to Account actions.
* **Code Table**—Stores texts that are required by Finance Service for framework and business logic implementations.
* **CodeType Table**—Stores data that is related to a logical group identified in the CodeType Table.
* **Refund Queue Fee Table**—Stores detailed data about a refund.
* **AriaUser**—Stores data the UserID generated for Aria.
* **PaymentGatewayRequestStatus Table**—Stores the ClientHashValue and its status.

**Also See**

“A.2.1 Asurion Finance Database Tables” in the “Asurion Subscriber System Product Specification.”

## CRM Database

The CRM Database stores the requests and responses to manually entered enrollment, contact, billing cycle, invoice, and other information about Finance Service customer account. The CalculateTax API calls the CRM Web Service to update the DB. The database includes the following tables:

* **AncillaryData Table**—Stores ancillary data about contacts.
* **ValidationSummary Table**—Stores validation summary data.
* **asu\_enrollment Table**— Stores overview information about policy enrollments.
* **Contact Table**—Stores contact information.
* **asu\_billingcycle Table**—Stores billing cycle information.
* **asu\_communicationdetail Table** —Stores detailed information about communications.
* Error! Reference source not found.—Stores information about communication methods.
* Error! Reference source not found.—Stores information about system configurations.
* **asu\_configurationitems Table**—Stores configuration items (for example, “asur\_configurationid”).
* **asu\_contentgroup Table**— Stores content group information.
* **asu\_contentscript Table**—Stores content scripts for each account.
* **asu\_contentscriptset Table**—Stores content scripts
* Error! Reference source not found.—Stores logs of all manual requests and responses.
* **asu\_culture Table**—Stores culture information used by GatewayAPIs.
* Error! Reference source not found.—Stores information about holidays celebrated by supported cultures.
* Error! Reference source not found.—Stores information about payment methods.
* **asu\_program Table**—Stores information about programs.
* **Invoice Table**—Stores information about invoices.

**Also See**

“A.2.2 CRM Database Tables” in the “Asurion Subscriber System Product Specification.”

# Appendix

## A.1 Asurion Finance Service Gateway API Description

The following sections present the XML schema for the Asurion Finance Service APIs: CreateContract Service, ProcessNonRefPayment Service, and CalculateTaxes Service.

### A.1.1 CreateContract Service API

The CreateContractService API has message request and response components. The XML schemas for these elements are shown below.

#### CreateContractRequest XML Schema

The XML schema for the API is shown below.

<?xml version="1.0" encoding="UTF-8"?>

<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"

xmlns:ns1="http://services.asurion.com/schemas/PolicyAdministration/1.0.0"

xmlns:tns="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0"

xmlns="http://services.asurion.com/schemas/Finance/CreateContractsRequest/1.1"

targetNamespace="http://services.asurion.com/schemas/Finance/CreateContractsRequest/1.1"

elementFormDefault="qualified"

attributeFormDefault="unqualified">

<s:import namespace="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0" schemaLocation="AsurionCanonicalModel.xsd"/>

<s:element name="CreateContractsRequest" type="CreateContractsRequest"/>

<s:complexType name="CreateContractsRequest">

<s:complexContent>

<s:extension base="tns:BaseRequest">

<s:sequence>

<s:element name="Contract" type="Contract" maxOccurs="unbounded"/>

</s:sequence>

</s:extension>

</s:complexContent>

</s:complexType>

<s:complexType name="Contract">

<s:sequence>

<s:element name="ContractHolder" type="Contact"/>

<s:element name="FeatureCode" type="s:string"/>

<s:element name="PremiumPaymentTransactionId" type="s:string"/>

<s:element name="ContractStatus" type="s:string"/>

<s:element name="ContractStatusReason" type="s:string" minOccurs="0"/>

<s:element name="AncillaryData" type="AncillaryData" minOccurs="0" maxOccurs="unbounded"/>

<s:element name="ContractedProducts" type="ArrayOfContractedProducts"/>

<s:element name="EnrollmentDate" type="s:dateTime"/>

<s:element name="DateOfBirth" type="s:string" minOccurs="0"/>

<s:element name="IsContractValidationDone" type="s:boolean"/>

<s:element name="IsAddressStandardized" type="s:boolean"/>

</s:sequence>

</s:complexType>

<s:complexType name="AncillaryData">

<s:sequence>

<s:element name="Name" type="s:string"/>

<s:element name="Value" type="s:string"/>

<s:element name="Type" type="s:string"/>

<s:element name="Sequence" type="s:string"/>

</s:sequence>

</s:complexType>

<s:complexType name="ArrayOfContractedProducts">

<s:sequence>

<s:element name="ContractedProduct" type="Product" minOccurs="0" maxOccurs="unbounded"/>

</s:sequence>

</s:complexType>

<s:complexType name="Product">

<s:sequence>

<s:element name="Make" type="s:string"/>

<s:element name="Model" type="s:string"/>

<s:element name="DeviceId" type="s:string"/>

<s:element name="Owner" type="Contact"/>

<s:element name="PurchaseDate" type="s:dateTime" minOccurs="0"/>

</s:sequence>

</s:complexType>

<s:complexType name="Address">

<s:sequence>

<s:element name="Address1" type="s:string"/>

<s:element name="Address2" type="s:string" minOccurs="0"/>

<s:element name="City" type="s:string"/>

<s:element name="Region" type="s:string"/>

<s:element name="PostalCode" type="s:string"/>

<s:element name="Country" type="s:string"/>

</s:sequence>

</s:complexType>

<s:complexType name="Contact">

<s:sequence>

<s:element name="FirstName" type="s:string"/>

<s:element name="MiddleName" type="s:string" minOccurs="0"/>

<s:element name="LastName" type="s:string"/>

<s:element name="Salutation" type="s:string" minOccurs="0"/>

<s:element name="EMail" type="s:string" minOccurs="0"/>

<s:element name="Telephone" type="s:string"/>

<s:element name="Address1" type="Address"/>

</s:sequence>

</s:complexType>

</s:schema>

#### CreateContractResponse XML Schema

The XML schema for the API is shown below.

<?xml version="1.0" encoding="UTF-8"?>

<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"

xmlns:ns1="http://services.asurion.com/schemas/PolicyAdministration/1.0.0"

xmlns:tns="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0"

xmlns="http://services.asurion.com/schemas/Finance/CreateContractsResponse/1.1"

targetNamespace="http://services.asurion.com/schemas/Finance/CreateContractsResponse/1.1"

elementFormDefault="qualified"

attributeFormDefault="unqualified">

<s:import namespace="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0" schemaLocation="AsurionCanonicalModel.xsd"/>

<s:element name="CreateContractsResponse" type="CreateContractsResponse"/>

<s:complexType name="CreateContractsResponse">

<s:complexContent>

<s:extension base="tns:BaseResponse">

<s:sequence>

<s:element name="Response" type="s:string"/>

</s:sequence>

</s:extension>

</s:complexContent>

</s:complexType>

</s:schema>.

### A.1.2 ProcessNonRefPayment Service API

The ProcessNonRefPaymentService API has request and response components. The XML schemas for these elements are shown below.

#### ProcessNonRefPaymentRequest XML Schema

The XML schema for the API is shown below.

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:tns="http://services.asurion.com/schemas/ProcessNonRefPaymentRequest/1.0.1"

xmlns:ns1="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0"

targetNamespace="http://services.asurion.com/schemas/ProcessNonRefPaymentRequest/1.0.1"

elementFormDefault="qualified"

attributeFormDefault="unqualified">

<xsd:import namespace="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0" schemaLocation="AsurionCanonicalModel.xsd"/>

<xsd:element name="ProcessNonRefPaymentRequest" type="tns:ProcessNonRefPayment"/>

<xsd:complexType name="ProcessNonRefPayment">

<xsd:complexContent>

<xsd:extension base="ns1:BaseRequest">

<xsd:sequence>

<xsd:element name="ProcessNonRefPaymentWithTender" type="tns:ProcessNonRefPaymentWithTender" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="ProcessNonRefPaymentWithTender">

<xsd:sequence>

<xsd:element name="BillingAddress" type="tns:BillingAddress" minOccurs="0"/>

<xsd:element name="Customer" type="tns:Customer"/>

<xsd:element name="Tender" type="tns:Tender"/>

<xsd:element name="DataKey" type="xsd:string"/>

<xsd:element name="TransactionType" type="xsd:string"/>

<xsd:element name="Comment1" type="xsd:string"/>

<xsd:element name="Comment2" type="xsd:string"/>

<xsd:element name="TransactionAmount" type="xsd:double"/>

<xsd:element name="CurrencyCode" type="xsd:string"/>

<xsd:element name="BatchOnFailure" type="xsd:boolean"/>

<xsd:element name="FeeType" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="BillingAddress">

<xsd:sequence>

<xsd:element name="AddressLine1" type="xsd:string"/>

<xsd:element name="AddressLine2" type="xsd:string" minOccurs="0"/>

<xsd:element name="AddressLine3" type="xsd:string" minOccurs="0"/>

<xsd:element name="City" type="xsd:string"/>

<xsd:element name="Region" type="xsd:string"/>

<xsd:element name="PostalCode" type="xsd:string"/>

<xsd:element name="Country" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Customer">

<xsd:sequence>

<xsd:element name="FirstName" type="xsd:string"/>

<xsd:element name="LastName" type="xsd:string"/>

<xsd:element name="EmailAddress" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Tender">

<xsd:choice>

<xsd:element name="CreditCardTender" type="tns:CreditCardTender" minOccurs="0"/>

</xsd:choice>

</xsd:complexType>

<xsd:complexType name="CreditCardTender">

<xsd:sequence>

<xsd:element name="AccountNumber" type="xsd:string"/>

<xsd:element name="CardType" type="xsd:string"/>

<xsd:element name="CVVCode" type="xsd:string"/>

<xsd:element name="ExpirationMonth" type="xsd:int"/>

<xsd:element name="ExpirationYear" type="xsd:int"/>

<xsd:element name="CheckCVVCode" type="xsd:boolean"/>

<xsd:element name="CheckAVS" type="xsd:boolean"/>

</xsd:sequence>

</xsd:complexType>

</xsd:schema>

#### ProcessNonRefPaymentResponse XML Schema

The XML schema for the API is shown below.

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:ns1="http://services.asurion.com/schemas/ProcessNonRefPaymentResponse/1.0.1"

xmlns:tns="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0"

targetNamespace="http://services.asurion.com/schemas/ProcessNonRefPaymentResponse/1.0.1"

elementFormDefault="qualified"

attributeFormDefault="unqualified">

<xsd:import namespace="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0" schemaLocation="AsurionCanonicalModel.xsd"/>

<xsd:element name="ProcessNonRefPaymentResponse" type="ns1:ProcessNonRefPaymentResult"/>

<xsd:complexType name="ProcessNonRefPaymentResult">

<xsd:complexContent>

<xsd:extension base="tns:BaseResponse">

<xsd:sequence>

<xsd:element name="ProcessedTransaction" type="ns1:ProcessedTransaction" maxOccurs="unbounded"/>

<xsd:element name="ClientHashValue" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="ProcessedTransaction">

<xsd:sequence>

<xsd:element name="Application" type="xsd:string" minOccurs="0"/>

<xsd:element name="PaymentMethod" type="xsd:string" minOccurs="0"/>

<xsd:element name="PaymentStatus" type="xsd:string" minOccurs="0"/>

<xsd:element name="TransactionDetails" type="ns1:BillingAddress" minOccurs="0"/>

<xsd:element name="Caller" type="ns1:Caller" minOccurs="0"/>

<xsd:element name="GatewayDetail" type="ns1:GatewayDetail" minOccurs="0"/>

<xsd:element name="IsSuccessful" type="xsd:string" minOccurs="0"/>

<xsd:element name="ReferenceNumber" type="xsd:string" minOccurs="0"/>

<xsd:element name="Request" type="xsd:string" minOccurs="0"/>

<xsd:element name="Response" type="xsd:string" minOccurs="0"/>

<xsd:element name="TenderDetail" type="ns1:TenderDetail" minOccurs="0"/>

<xsd:element name="TransactionAmount" type="xsd:double"/>

<xsd:element name="CurrencyCode" type="xsd:string" minOccurs="0"/>

<xsd:element name="TransactionDetailId" type="xsd:long"/>

<xsd:element name="DataKey" type="xsd:string" minOccurs="0"/>

<xsd:element name="PaymentMethod" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="BillingAddress">

<xsd:sequence>

<xsd:element name="AddressLine1" type="xsd:string" minOccurs="0"/>

<xsd:element name="AddressLine2" type="xsd:string" minOccurs="0"/>

<xsd:element name="AddressLine3" type="xsd:string" minOccurs="0"/>

<xsd:element name="City" type="xsd:string" minOccurs="0"/>

<xsd:element name="Region" type="xsd:string" minOccurs="0"/>

<xsd:element name="PostalCode" type="xsd:string" minOccurs="0"/>

<xsd:element name="Country" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Caller">

<xsd:sequence>

<xsd:element name="FirstName" type="xsd:string" minOccurs="0"/>

<xsd:element name="LastName" type="xsd:string" minOccurs="0"/>

<xsd:element name="EmailAddress" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="GatewayDetail">

<xsd:sequence>

<xsd:element name="Comment1" type="xsd:string" minOccurs="0"/>

<xsd:element name="Comment2" type="xsd:string" minOccurs="0"/>

<xsd:element name="TransactionDetailId" type="xsd:int"/>

<xsd:element name="TransType" type="xsd:string" minOccurs="0"/>

<xsd:element name="AuthCode" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="TenderDetail">

<xsd:choice>

<xsd:element name="CreditCardTender" type="ns1:CreditCardTender" minOccurs="0"/>

</xsd:choice>

</xsd:complexType>

<xsd:complexType name="CreditCardTender">

<xsd:sequence>

<xsd:element name="AccountNumber" type="xsd:string" minOccurs="0"/>

<xsd:element name="CardType" type="xsd:string" minOccurs="0"/>

<xsd:element name="ExpirationMonth" type="xsd:int"/>

<xsd:element name="ExpirationYear" type="xsd:int"/>

</xsd:sequence>

</xsd:complexType>

</xsd:schema>

### A.1.3 CalculateTaxes Service API

The CalculateTaxes Service API has request and response components. The XML schemas for these elements are shown below.

#### CalculateTaxesRequest

The XML schema for the API is shown below.

<?xml version="1.0" encoding="UTF-8"?>

<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"

xmlns:ns1="http://services.asurion.com/schemas/Finance/1.0.0"

xmlns="http://services.asurion.com/schemas/Finance/CalculateTaxesRequest/1.0"

xmlns:tns="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0"

targetNamespace="http://services.asurion.com/schemas/Finance/CalculateTaxesRequest/1.0"

elementFormDefault="qualified"

attributeFormDefault="unqualified">

<s:import namespace="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0" schemaLocation="AsurionCanonicalModel.xsd"/>

<!--s:import namespace="http://services.asurion.com/schemas/Finance/1.0.0" schemaLocation="s/Common/FinanceModel.xsd"/-->

<s:element name="CalculateTaxesRequest" type="CalculateTaxesRequest"/>

<s:complexType name="CalculateTaxesRequest">

<s:complexContent>

<s:extension base="tns:BaseRequest">

<s:sequence>

<s:element name="CalculateTaxes" type="CalculateTaxes" maxOccurs="unbounded"/>

</s:sequence>

</s:extension>

</s:complexContent>

</s:complexType>

<s:complexType name="CalculateTaxes">

<s:sequence>

<s:element name="invoiceNumber" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="invoiceAmount" type="s:string"/>

<s:element name="invoiceType" type="s:string"/>

<s:element name="forEstimateOnly" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="countryCode" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="shipToAddress" type="Address"/>

<s:element name="shipFromAddress" type="Address" nillable="true" minOccurs="0"/>

<s:element name="orderToAddress" type="Address" nillable="true" minOccurs="0"/>

</s:sequence>

</s:complexType>

<s:complexType name="Address">

<s:sequence>

<s:element name="addressLine1" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="addressLine2" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="city" type="s:string"/>

<s:element name="state" type="s:string"/>

<s:element name="postalCode" type="s:string"/>

<s:element name="country" type="s:string" nillable="true" minOccurs="0"/>

</s:sequence>

</s:complexType>

</s:schema>

#### CalculateTaxesResponse

The XML schema for the API is shown below.

<?xml version="1.0" encoding="UTF-8"?>

<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"

xmlns:s1="http://services.asurion.com/schemas/Finance/1.0.0"

xmlns="http://services.asurion.com/schemas/Finance/CalculateTaxesResponse/1.1"

xmlns:tns="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0"

targetNamespace="http://services.asurion.com/schemas/Finance/CalculateTaxesResponse/1.1"

elementFormDefault="qualified"

attributeFormDefault="unqualified">

<s:import namespace="http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0" schemaLocation="AsurionCanonicalModel.xsd"/>

<!--s:import namespace="http://services.asurion.com/schemas/Finance/1.0.0" schemaLocation="../../../SharedResources/Schemas/Common/FinanceModel.xsd"/-->

<s:element name="CalculateTaxesResponse" type="CalculateTaxesResponse"/>

<s:complexType name="CalculateTaxesResponse">

<s:complexContent>

<s:extension base="tns:BaseResponse">

<s:sequence>

<s:element name="CalculateTaxesResult" type="CalculateTaxesResult" maxOccurs="unbounded"/>

</s:sequence>

</s:extension>

</s:complexContent>

</s:complexType>

<s:complexType name="CalculateTaxesResult">

<s:sequence>

<s:element name="returnCode" type="s:string"/>

<s:element name="returnMessage" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="invoiceNumber" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="totalTaxAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="totalTaxRate" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="taxDetail" type="DetailedTaxResultType" nillable="true" minOccurs="0"/>

</s:sequence>

</s:complexType>

<s:complexType name="DetailedTaxResultType">

<s:sequence>

<s:element name="detailedUSTax" type="DetailedUSTaxResultType" nillable="true" minOccurs="0"/>

<s:element name="detailedCanadaTax" type="CanadianTaxResult" nillable="true" minOccurs="0"/>

</s:sequence>

</s:complexType>

<s:complexType name="DetailedUSTaxResultType">

<s:sequence>

<s:element name="stateTaxAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="stateTaxRate" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="countyTaxAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="countyTaxRate" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="districtTaxAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="districtTaxRate" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="cityTaxAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="cityTaxRate" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="countyName" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="geocodeValue" type="s:string" nillable="true" minOccurs="0"/>

</s:sequence>

</s:complexType>

<s:complexType name="CanadianTaxResult">

<s:sequence>

<s:element name="PSTAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="QSTAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="GSTAmount" type="s:string" nillable="true" minOccurs="0"/>

<s:element name="HSTAmount" type="s:string" nillable="true" minOccurs="0"/>

</s:sequence>

</s:complexType>

</s:schema>

## A.3 Example: CreateContact Request to Create a Billing Account

<ns0:BaseRequest xmlns:ns0="<http://services.asurion.com/wsdl/1.0/Finance/1.0>">

    <body>

        <BaseRequest xmlns:q1="<http://services.asurion.com/schemas/Finance/CreateContractsRequest/1.1>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" xmlns:SOAP-ENV="<http://schemas.xmlsoap.org/soap/envelope/>" xmlns:s="<http://schemas.xmlsoap.org/soap/envelope/>" xmlns="<http://services.asurion.com/schemas/AsurionCanonicalModel/2.0.0>" xsi:type="q1:CreateContractsRequest">

            <q1:Contract>

                <q1:ContractHolder>

                    <q1:FirstName>ASURION</q1:FirstName>

                    <q1:MiddleName />

                    <q1:LastName>PRODTEST4</q1:LastName>

                    <q1:Salutation />

                    <q1:EMail>[test@gmail.com</q1:EMail](mailto:test@gmail.com%3c/q1:EMail)>

                    <q1:Telephone>8005007474</q1:Telephone>

                    <q1:Address1>

                        <q1:Address1>805 MOBERLY LN</q1:Address1>

                        <q1:Address2 />

                        <q1:City>BENTONVILLE</q1:City>

                        <q1:Region>AR</q1:Region>

                        <q1:PostalCode>72712</q1:PostalCode>

                        <q1:Country>USA</q1:Country>

                    </q1:Address1>

                </q1:ContractHolder>

                <q1:FeatureCode>10081754</q1:FeatureCode>

                <q1:PremiumPaymentTransactionId>101700238</q1:PremiumPaymentTransactionId>

                <q1:ContractStatus>INITIAL</q1:ContractStatus>

                <q1:ContractStatusReason>Customer</q1:ContractStatusReason>

                <q1:AncillaryData>

                    <q1:Name>Store Code</q1:Name>

                    <q1:Value>02420</q1:Value>

                    <q1:Type>String</q1:Type>

                    <q1:Sequence>16</q1:Sequence>

                </q1:AncillaryData>

                <q1:AncillaryData>

                    <q1:Name>Wireless Carrier</q1:Name>

                    <q1:Value>AT&amp;T</q1:Value>

                    <q1:Type>String</q1:Type>

                    <q1:Sequence>17</q1:Sequence>

                </q1:AncillaryData>

                <q1:AncillaryData>

                    <q1:Name>Wireless Carrier Account Number</q1:Name>

                    <q1:Value>1234134</q1:Value>

                    <q1:Type>String</q1:Type>

                    <q1:Sequence>18</q1:Sequence>

                </q1:AncillaryData>

                <q1:AncillaryData>

                    <q1:Name>Type Of Equipment</q1:Name>

                    <q1:Value>C</q1:Value>

                    <q1:Type>String</q1:Type>

                    <q1:Sequence>19</q1:Sequence>

                </q1:AncillaryData>

                <q1:ContractedProducts>

                    <q1:ContractedProduct>

                        <q1:Make>HTC</q1:Make>

                        <q1:Model>FREESTLGRY</q1:Model>

                        <q1:DeviceId>123444444444444</q1:DeviceId>

                        <q1:Owner>

                            <q1:FirstName>ASURION</q1:FirstName>

                            <q1:MiddleName />

                            <q1:LastName>PRODTEST4</q1:LastName>

                            <q1:Salutation />

                            <q1:EMail>[test@gmail.com</q1:EMail](mailto:test@gmail.com%3c/q1:EMail)>

                            <q1:Telephone>8005007474</q1:Telephone>

                            <q1:Address1>

                                <q1:Address1>805 MOBERLY LN</q1:Address1>

                                <q1:City>BENTONVILLE</q1:City>

                                <q1:Region>AR</q1:Region>

                                <q1:PostalCode>72712</q1:PostalCode>

                                <q1:Country>USA</q1:Country>

                            </q1:Address1>

                        </q1:Owner>

                        <q1:PurchaseDate>2011-10-26T08:53:28.9615829-05:00</q1:PurchaseDate>

                    </q1:ContractedProduct>

                </q1:ContractedProducts>

                <q1:EnrollmentDate>2011-10-26T08:53:28.9615829-05:00</q1:EnrollmentDate>

                <q1:DateOfBirth>10101981</q1:DateOfBirth>

                <q1:IsContractValidationDone>true</q1:IsContractValidationDone>

                <q1:IsAddressStandardized>true</q1:IsAddressStandardized>

            </q1:Contract>

        </BaseRequest>

    </body>

</ns0:BaseRequest>

# Index

ARIA, 7, 10

Asurion.PRODNA.Q.Business.Persistent.Financ e.Enterprise.InsertBehaviorStateMDM, 9

Asurion.PRODNA.Q.Business.Persistent.Finance.CreateContract, 9

Asurion.PRODNA.Q.Business.Persistent.Finance.ProcessGatewayMessage, 8

Asurion.PRODNA.Q.Business.Persistent.Finance.ProcessNonRefPayment, 11

AsurionFinance Database Table

AriaUser, 16

BTADetail, 16

CreditCardDetail, 16

DeuCSDetail, 16

DirectDebitDetail, 16

ECheckDetail, 16

EDIFactDetail, 16

InStoreDetail, 16

RefundQueue, 16

RefundQueueFee, 16

TransactionDetail, 16

AsurionFinance Database Table

Transaction, 16

AsurionFinance Database VerisignDetail, 16

BaseRequest, 7, 13, 18, 20, 22, 25, 26

BaseResponse, 7, 13, 19, 21, 23

BusinessWorks (BW), 5

CalculateTaxes, 5, 8, 12, 13, 14, 15, 16, 18, 22, 23

CalculateTaxes API, 15

CalculateTaxesGateway, 5, 6, 8, 12

CreateContract, 5, 6, 8, 9, 10, 13, 14, 18

CreateContract API, 14

CreateContractRequest, 14, 18

CreateContractResponse, 14, 19

CreateTaxesGateway, 5

CRM, 5, 7, 9, 10, 16, 17

CRM Database, 17

CRM Database Table

AncillaryData, 17

Asu\_billingcycle, 17

Asu\_communicationmethod, 17

Asu\_configuration, 17

Asu\_configurationitems, 17

Asu\_contentscriptset, 17

Asu\_culture, 17

Asu\_emsactivitylogger, 17

Asu\_enrollment, 17

Asu\_holiday, 17

Asu\_program, 17

Asupaymethod, 17

Contact, 17

Invoice, 17

ValidationSummary, 17

Enterprise Canonical Model, 12

Enterprise SOAP Message Log, 9, 10

ESB, 9, 11, 12, 16

Finance Service Gateway, 14

Finance Services Gateway, 5

FinanceGateway, 5

FinanceService WSDL, 12

FinanceServiceAgent, 5, 7

FSA, 7, 8, 9, 11, 12, 14, 15

Gateway, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18

GetContracts, 5, 7, 10

Global Collect, 10, 11

HTTP, 7, 9, 10, 11, 12

InsertBehaviorStateMDM, 9, 10

ProcessNonRefPayment, 5, 6, 8, 11, 13, 15, 18, 20

ProcessNonRefPayment API, 15

ProcessNonRefPaymentGateway, 5, 6, 8, 11

ProcessNonRefPaymentRequest, 20

ProcessNonRefPaymentResponse, 21

ProcessNonRefPremiumPayment, 10, 11

Proxy Server, 7, 10

SOAP, 9, 10, 11, 12, 14, 15, 16, 25

Tibco, 5, 14

TIBCO XML WSDL, 13

Vertex, 12

VIP, 7

Virtual IP, 7

WalMart, 14

WSDL, 13

XML schema, 12, 18, 19, 20, 21, 22, 23