



Avalara AvaTax for Salesforce B2C Commerce

SiteGenesis JavaScript Controllers (SGJC)

System Integration Document for Avalara AvaTax

Revision Date: 03/06/2020

Product Release: v20.1.0

Avalara may have patents, patent applications, trademarks, copyrights, or other intellectual property rights governing the subject matter in this document. Except as expressly provided in any written license agreement from Avalara, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2018, 2019, 2020 Avalara, Inc. All rights reserved.

Avalara, AvaTax, CertCapture are either registered trademarks or trademarks of Avalara, Inc. in the United States or other countries.

All other trademarks are property of their respective owners.

Table of Contents

1.0. Acronyms and Abbreviations	3
2.0. Connector Overview	4
3.0. Configuration	4
4.0. Time Zone Differences.....	5
5.0. Object Mapping.....	6
5.1. Address Validation	6
5.2. Tax Calculation.....	6
6.0. Integration Points.....	9
7.0. Services.....	9
8.0. Workflow Diagrams.....	10

1.0. Acronyms and Abbreviations

B2C	Business to Consumer
SGJC	SiteGenesis JavaScript Controllers
UUID	Universally Unique Identifiers

2.0. Connector Overview

The AvaTax Salesforce B2C Commerce – SGJC cartridge provides rapid integration for Salesforce B2C Commerce – SGJC implementations. The AvaTax cartridge is a self-contained package that can easily integrate into a project. This cartridge can be configured in the Business Manager and contains all the elements necessary to perform a successful implementation of AvaTax **Tax calculation** and **Address Validation** services.

3.0. Configuration

The following table displays the list of AvaTax Configuration Settings:

Sr. No.	AvaTax Settings	Descriptions
Account Settings		
1.	Tax Calculation	<ul style="list-style-type: none"> • Enable AvaTax Tax Calculation on the transactions.
2.	Address Validation	<ul style="list-style-type: none"> • Enable Address Validation using AvaTax.
3.	Save Transaction to AvaTax	<ul style="list-style-type: none"> • Enable this to save the transactions to AvaTax.
4.	Commit Transaction to AvaTax	<ul style="list-style-type: none"> • Enable this to commit the transactions to AvaTax.
5.	Company Code	<ul style="list-style-type: none"> • The entered Company Code will be used to save the transactions in the company configured in the AvaTax.
6.	Customer Code	<ul style="list-style-type: none"> • Customer Code sent to Avalara services can be customized in the AvaTax Settings section. • For Guest users, Email ID provided during the checkout process is sent over. If the customer is authenticated, and the custom attribute preference value is available, then the attribute value from 'Profile' object is used. • The order of preference used for Customer Code is: <ul style="list-style-type: none"> ○ Custom field attribute provided in AvaTax Settings section for the Authenticated users and subject to the availability. ○ Email ID provided during checkout for both Authenticated and Guest Users. ○ Email ID from the User Account (authenticated users only).
7.	Shipping Tax Code	<ul style="list-style-type: none"> • The default Tax Code is used when no tax code is assigned to the shipping method.
Cross Border Calculation		
8.	Taxation Policy	<ul style="list-style-type: none"> • It shows the taxation policy set by the merchant.

Sr. No.	AvaTax Settings	Descriptions
Address Details		
1.	Location Code	<ul style="list-style-type: none"> If a value is available, then the entered origin address details are not used for the tax calculation.
2.	ShipFrom Address fields (Address Line 1, Address Line 2, Address Line 3, City, State, Zip/Postal Code, Country Code)	<ul style="list-style-type: none"> Enter the ShipFrom address details.

4.0. Time Zone Differences

The transaction date and tax date may vary if the customer performs the transaction in the different time zone than that of the AvaTax Servers.

The following steps allows you to resolve the time zone difference:

1. Navigate to `int_avatax > cartridge > scripts > avatax > avautil > avaTax.js`,
2. Find the 'getFormattedDate' function, and modify the below methods:
 - a) `'date.getFullYear().toString()'` to `'date.getUTCFullYear().toString()'`
 - b) `'this.insertLeadingZero(date.getMonth() + 1) to 'this.insertLeadingZero(date.getUTCMonth() + 1)'`
 - c) `'this.insertLeadingZero(date.getDate() + 1) to 'this.insertLeadingZero(date.getUTCDate() + 1)'`

5.0. Object Mapping

The following tables shows the mapping of Salesforce B2C Commerce Fields with the AvaTax Fields:

5.1. Address Validation

Sr. No.	Type of Salesforce B2C Commerce Object	Salesforce B2C Commerce Fields	AvaTax Fields
1	String	address1	line1
2	String	address2	line2
3	String	address3	line3
4	String	city	city
5	String	stateCode	region
6	String	countryCode / country	country
7	String	postalCode	postalCode
8	String	latitude	latitude
9	String	longitude	longitude

5.2. Tax Calculation

Sr. No.	Type of Salesforce B2C Commerce Object	Salesforce B2C Commerce Fields	AvaTax Fields	Descriptions
1.	String	Basket > UUID	code	<ul style="list-style-type: none"> Once you place an order, the variable code gets assign with the Order Number, and the docType gets assign with SalesInvoice.
2.	String	Settings > Company Code	companyCode	<ul style="list-style-type: none"> A custom field for the company code is provided. If the user enters the company code in this field, then the transaction passes to the mapped company code. If a company code in the custom field is blank, then the transaction passes to the Company Code mapped in the AvaTax configuration page.
3.	String	Current date	date	<ul style="list-style-type: none"> Retrieved using JavaScript.
4.	String	Basket > Shipment > Email	customerCode	<ul style="list-style-type: none"> It sends the Account Id and Name as a Customer Code.
5.	dw.order.orderAddress	Basket > Shipment > shippingAddress	addresses.shipTo	<ul style="list-style-type: none"> The shipTo address picked from shippingAddress and passes as destination address at header level address.

Sr. No.	Type of Salesforce B2C Commerce Object	Salesforce B2C Commerce Fields	AvaTax Fields	Descriptions
6.	dw.order.orderAddress	Settings > shipFrom values	addresses.shipFrom	<ul style="list-style-type: none"> The shipFrom address picked from the custom preferences and passes as an origin address at header level address.
7.	String	Shipment > Line Item > UUID	line.number	<ul style="list-style-type: none"> Represents one-line item in a transaction.
8.	Number	Line Item > Quantity	line.quantity	<ul style="list-style-type: none"> Quantity of items in the line.
9.	dw.value.Money	Line Item > Price	line.amount	<ul style="list-style-type: none"> The AvaTax receives the final amount once the discount or coupon applied to the current line item.
10.	dw.order.orderAddress	Settings > shipFrom values	line.addresses.shipFrom	<ul style="list-style-type: none"> The shipFrom address picked from the custom preferences and passes as a line level address.
11.	dw.order.orderAddress	Basket > Shipment > shippingAddress	line.addresses.shipTo	<ul style="list-style-type: none"> The shipTo address is filled by the end user on the Storefront and passes as a line level address.
12.	String	Line Item > Tax class ID OR Setting > defaultShippingMethodTaxCode	line.taxCode	<ul style="list-style-type: none"> If it is a product then the tax class ID configured for the product is used. If it is a shipping line item the tax code configured in custom preferences is used.
13.	dw.order.TaxMgr.taxationPolicy	Site settings > Taxation policy	line.taxIncluded	<ul style="list-style-type: none"> It is a taxation policy of the current site that is net/ gross.
14.	String	Line Item > ID	line.itemCode	<ul style="list-style-type: none"> The Line Item ID is used as ItemCode.
15.	String	Line Item > Product > Description	line.description	<ul style="list-style-type: none"> AvaTax for Salesforce B2C Commerce passes Product Description as a line description.
16.	Boolean	customer.profile.customer.isSellerImporterOfRecord	isSellerImporterOfRecord	<ul style="list-style-type: none"> Specifies if the transaction should have value-added and cross-border taxes calculated with the seller as the importer of record.
17.	String	customer.profile.customer.taxID	businessIdentificationNo	<ul style="list-style-type: none"> VAT business identification number for the customer for this transaction.

Sr. No.	Type of Salesforce B2C Commerce Object	Salesforce B2C Commerce Fields	AvaTax Fields	Descriptions
18.	String	basket.currencyCode	currencyCode	<ul style="list-style-type: none"> It is a currency code. For example: Currency Code: USD
19.	String	basket.custom.ATInvoiceMessage	response.messages.details[]	<ul style="list-style-type: none"> This field is used to display the invoice message received from AvaTax service.
20.	String	order.custom.ATInvoiceMessage	response.messages.details[]	<ul style="list-style-type: none"> This field is used to display the invoice message received from AvaTax service.
21.	Text	basket.custom.ATTaxDetail	response.line.details[]	<ul style="list-style-type: none"> It stores the tax detail/ type in the basket object.
22.	Text	order.custom.ATTaxDetail	response.line.details[]	<ul style="list-style-type: none"> It stores the tax detail/ type in the order object.
23.	String	basket.custom.ATCustomDuty	response.summary[i].tax	<ul style="list-style-type: none"> It is used to store import duty/ fees if applicable in the basket object.
24.	String	order.custom.ATCustomDuty	response.summary[i].tax	<ul style="list-style-type: none"> It is used to store import duty/ fees if applicable in the order object.
25.	Text	basket.custom.ATGenericMessage	response.messages[i].summary	<ul style="list-style-type: none"> It is used to store the transaction summary messages in the basket object.
26.	Text	order.custom.ATGenericMessage	response.messages[i].summary	<ul style="list-style-type: none"> It is used to store the transaction summary messages in the order object.
27.	Text	basket.custom.ATLandedCost	response.messages[i].summary	<ul style="list-style-type: none"> It is used to store Cross Border messages in the basket object.
28.	Text	order.custom.ATLandedCost	response.messages[i].summary	<ul style="list-style-type: none"> It is used to store Cross Border messages in the order object.
29.	String	basket.custom.ATTax	response.summary[i].tax	<ul style="list-style-type: none"> It is used to store tax.
30.	String	order.custom.ATTax	response.summary[i].tax	<ul style="list-style-type: none"> It is used to store tax.

6.0. Integration Points

The following table shows the AvaTax Integration Points:

SR. No.	Integration Page	Address Validation	Tax Calculation
1.	Shipping Page	Yes	Yes
2.	Billing Page	No	Yes

7.0. Services

Navigate to **Merchant Tools > AvaTax > AvaTax settings > Services**

- Test Connection – It is used to validate the AvaTax Account credentials.
- Void Transaction – It is used to void/cancel the orders posted in the AvaTax. The AvaTax service response is returned and displayed in the Service Response box.
- Commit Transaction It is used to commit the orders posted in the AvaTax. The AvaTax service response is returned and displayed in the Service Response box.
- Validate Address – It is used to validate shipping address of an order and receive a validated address as response.

Note: All the above features are for the testing purpose and can be scaled to process multiple Orders on the storefront.

8.0. Workflow Diagrams

Address Validation

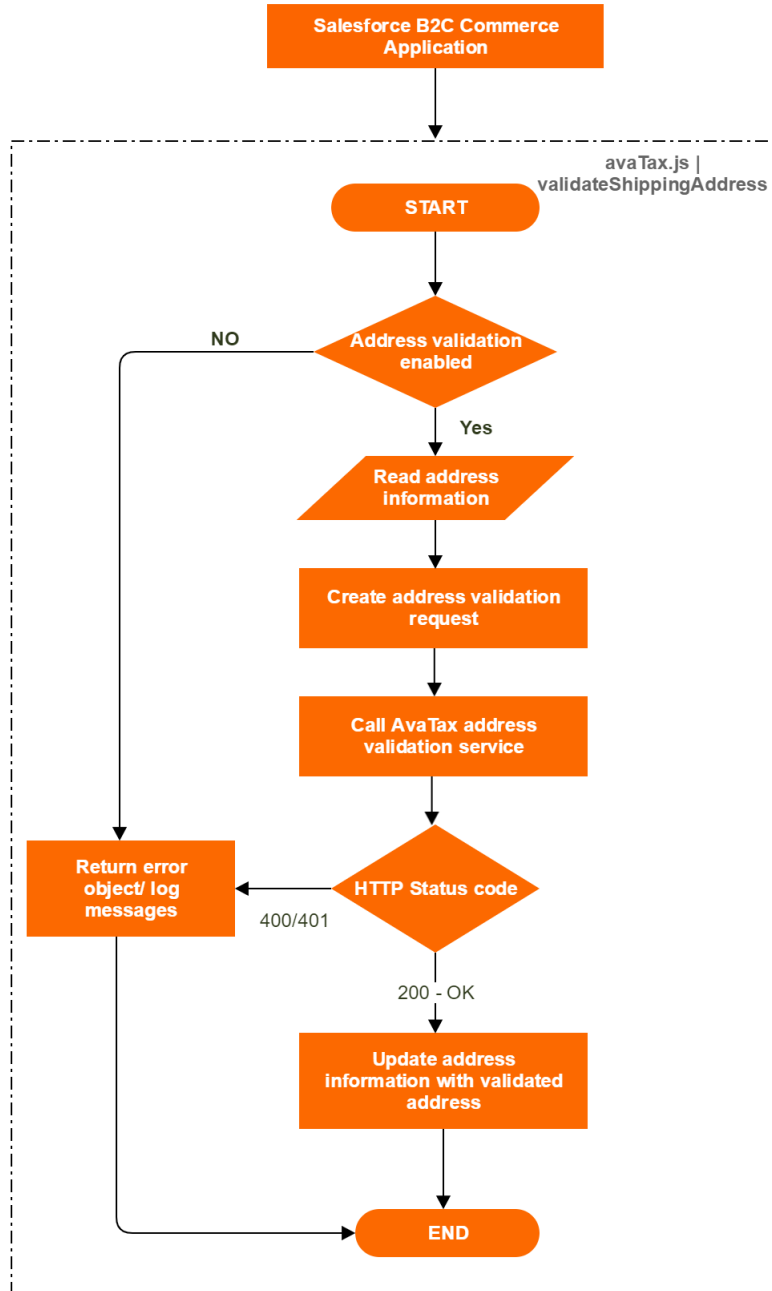


Figure 1: Address Validation

Tax Calculation

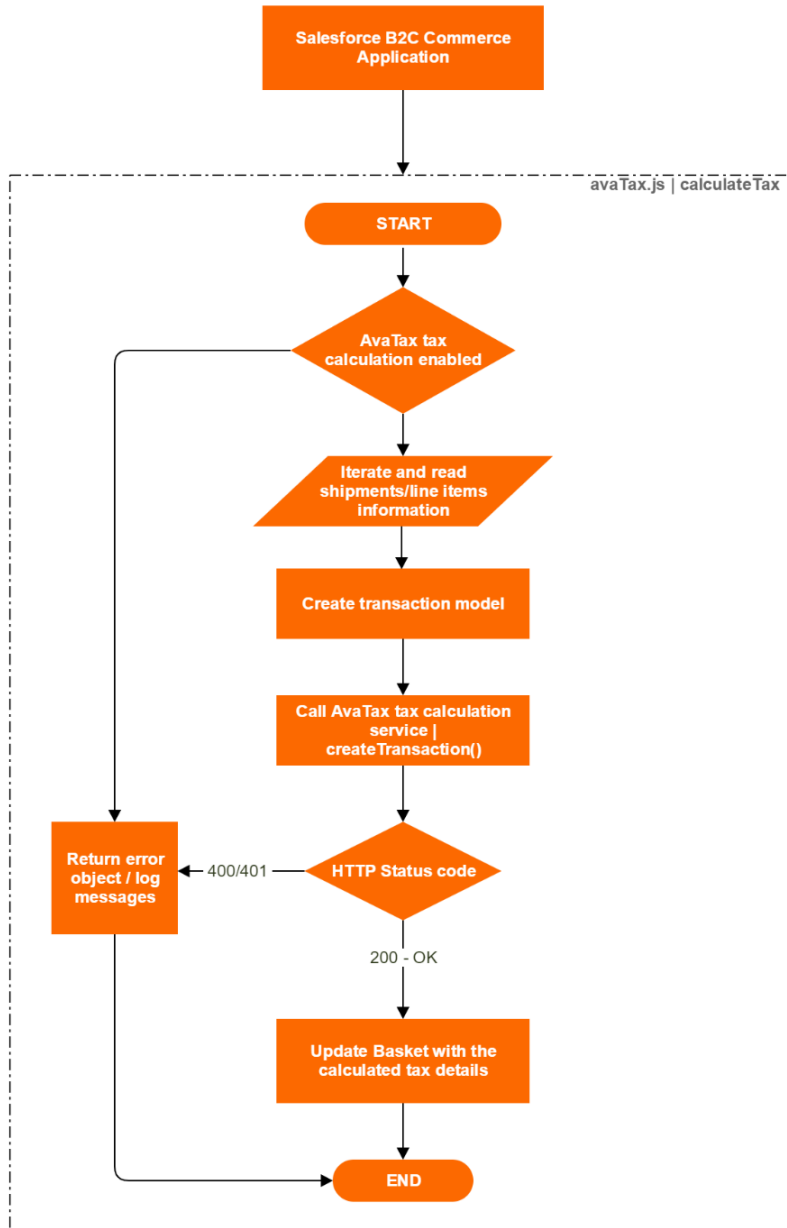


Figure 2: Tax Calculation