



# Avalara AvaTax for Salesforce B2C Commerce

## Storefront Reference Architecture (SFRA)

---

### User Guide

Revision Date: 12/09/2022

Product Release: 22.4.0

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, and 2022 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

<b>1.0. Summary</b>	<b>3</b>
<b>2.0. Component Overview</b>	<b>3</b>
2.1. Functional Overview	3
2.2. Use Cases	3
2.3. Limitations and Constraints	3
2.4. Compatibility	3
2.5. Privacy and Payment	3
<b>3.0. Implementation Guide</b>	<b>4</b>
3.1. Setup	4
3.1.1. Import Cartridges	4
3.1.2. Import Metadata	4
3.1.3. AvaTax Business Manager Extension Cartridge Setup	5
3.2. Configuration	5
3.2.1. Configure AvaTax Services	5
3.2.2. AvaTax Settings Configuration	7
3.3. Cross Border, VAT, and Invoice Messages	8
3.4. Services	12
3.5. Line Level Addressing	13
3.6. Beverages and Alcohol Tax Calculation	14
3.7. External Interfaces	19
3.8. Firewall Requirements	19
3.9. Testing	19
<b>4.0. Operations, Maintenance</b>	<b>20</b>
4.1. Data Storage	20
4.2. Availability	20
4.3. Support	20
4.4. Object Mapping	20
4.5. Object Mapping Documentation	20
<b>5.0. FAQs</b>	<b>21</b>
<b>6.0. Known Issues</b>	<b>23</b>
<b>7.0. Release History</b>	<b>24</b>

## 1.0. Summary

The AvaTax cartridge provides rapid integration for Salesforce B2C Commerce implementations. The AvaTax cartridge is a self-contained cartridge that can easily integrate into any project. This cartridge can be configured in the Business Manager and contains all elements necessary to perform a successful best practices implementation of AvaTax.

## 2.0. Component Overview

### 2.1. Functional Overview

1. AvaTax Update is a powerful, online sales tax compliance solution that provides all businesses with a level of sales tax automation that until now has been available only to Fortune 500 companies. AvaTax Update's transaction-based service model and seamless integration into business applications gives you rapid access to all jurisdiction assignments and real-time sales tax calculations. Cutting-edge technologies and superior processing logic can manage even the most complicated tax issues, such as situs, nexus, tax tiers, tax holidays, exemptions, certificate management and product taxability rules.
2. During check out shopping cart and customer information is transmitted via the AvaTax API web service. Tax information is sent back and applied to products in the cart.
3. Address Validation logic has been added to assist with filtering certain countries (configured in the custom site preferences) and to enable/disable the Address Validation calls.
4. Detailed transaction logging is available on the AvaTax Update dashboard.

### 2.2. Use Cases

This cartridge can be used to calculate taxes on the Salesforce B2C Commerce storefront through the AvaTax REST v2 API web service. It can also be used for address validation for all the US and Canada addresses.

Following features are provided as part of the cartridge:

1. Tax calculation
2. Address validation

### 2.3. Limitations and Constraints

As of now, only limitation is that AvaTax address validation is only available for US and Canada. AvaTax currently only supports service messages in English.

### 2.4. Compatibility

The latest release is tested with SFRA 6.0 and works with compatibility mode 22.7.

### 2.5. Privacy and Payment

Customer addresses are sent to and from AvaTax for Tax Calculation and Address Validation. No customer payment information is accessed or stored.

## 3.0. Implementation Guide

### 3.1. Setup

#### 3.1.1. Import Cartridges

Import `bm_avatax`, `int_avatax_sfra` and `int_avatax_svcclient` cartridges into the Eclipse UX Studio Workspace.

1. Open UX Studio in Eclipse IDE.
2. Go to **File > Import > General > Existing Projects into Workspace**
3. Browse to the directory where the cartridges have been downloaded.
4. Click **Finish**.
5. Click **OK**, when prompted to link the cartridges to the sandbox.
6. Modify the Site Path in **Business Manager > Administration > Manage Sites**. Make sure the cartridge names 'int\_avatax\_sfra' and 'int\_avatax\_svcclient' appear before any other cartridges.

If using Visual Studio code, use the below `dw.json` format to upload the cartridges to the sandbox. Place it at the root of the working directory.

```
{
  "hostname": "your-sandbox-hostname.demandware.net",
  "username": "yourlogin",
  "password": "yourpwd",
  "code-version": "version_to_upload_to"
}
```

#### 3.1.2. Import Metadata

1. Locate `site-import.zip` file inside metadata folder. Extract its content. Change site name if needed inside sites folder and compress the site-import folder to `site-import.zip` again.
2. Log in to the **Business Manager**.
3. Click **Administration > Site Development > Site Import & Export**.
4. Use the upload control to browse for the `site-import.zip` file located in the **metadata** folder.
5. Click **Upload**.
6. Select the `site-import.zip` and click **Import**. Import should complete successfully.

### 3.1.3. AvaTax Business Manager Extension Cartridge Setup

1. Make sure that the 'bm\_avatax' cartridge is imported in UX Studio workspace.
2. Add 'bm\_avatax', 'int\_avatax\_sfra' and 'int\_avatax\_svcclient' cartridges in BM cartridge path by going to – **Administration > Sites > Manage Sites > Business Manager –Settings**
3. Make sure that the AvaTax BM module has been added to Business Manager modules and has been assigned necessary permissions as follows –
  - Go to **Administration > Organization > Roles & Permissions**
  - Select the Role to give access to. e.g., Administrator
  - Go to **Business Manager Modules** section. Select the context for any or all the sites.
  - In the list of BM modules that are displayed, enable **AvaTax**. And click **Update**.
  - A new module **AvaTax** and its menu items should appear under **Merchant Tools**. Refer the below screenshot.



## 3.2. Configuration

### 3.2.1. Configure AvaTax Services

1. Make sure that the AvaTax services have been imported as part of site import step (3.1.2) performed above. Below service configurations should be present in the **Services** section.
2. Log in to the **Business Manager**.
3. Navigate to - **Administration > Operations > Services**
4. Configure your AvaTax credentials under **Credentials** tab, in `credentials.avatax.rest`.

Make sure that the service URL used is one of the following, depending on whether the sandbox is a production or development:

- **Development:** <https://sandbox-rest.avatax.com>
- **Production:** <https://rest.avatax.com>

5. Configure your AvaTax credentials under **Credentials** tab, in `logentries.avatax.rest`.

Make sure that the service URL used is one of the following, depending on whether the sandbox is a production or development:

- **Development:** <https://cepllogger.sbx.avalara.com/api/logger/>
- **Production:** <https://cepllogger.avalara.com/api/logger/>

### Notes:

- Services named `cepllogger.avatax.svc` and `avatax.rest.all` should also have been created with corresponding Profile and Credential items. **DO NOT remove these. Select the checkbox on 'Communication Log Enabled' in `cepllogger.avatax.svc` and `avatax.rest.all` as shown in the 'Service' screenshot below.**
- DO NOT modify service name(s) of any of the services.

Below are the screenshots for reference:

### Credentials:

<b>Name:</b> *	<input type="text" value="credentials.avatax.rest"/>
<b>URL:</b>	<input type="text" value="https://sandbox-rest.avatax.com/"/>
<b>User:</b>	<input type="text" value="xxxxxxxxxx"/>
<b>Password:</b>	<input type="password" value="....."/>

### Profile:

<b>Name:</b> *	<input type="text" value="profile.avatax.rest"/>
<b>Connection Timeout (ms):</b>	<input type="text" value="20,000"/>
<b>Enable Circuit Breaker:</b>	<input type="checkbox"/>
<b>Max Circuit Breaker Calls:</b>	<input type="text" value="0"/>
<b>Circuit Breaker Interval (ms):</b>	<input type="text" value="0"/>
<b>Enable Rate Limit:</b>	<input type="checkbox"/>
<b>Max Rate Limit Calls:</b>	<input type="text" value="0"/>
<b>Rate Limit Interval (ms):</b>	<input type="text" value="0"/>

**Service:**

<b>Name:</b> *	avatax.rest.all
<b>Type:</b>	HTTP
<b>Enabled:</b>	<input checked="" type="checkbox"/>
<b>Service Mode:</b>	Live
<b>Log Name Prefix:</b>	
<b>Communication Log Enabled:</b>	<input checked="" type="checkbox"/>
<b>Force PRD behavior in non-PRD environments:</b>	<input type="checkbox"/>
<b>Profile:</b>	profile.avatax.rest
<b>Credentials:</b>	credentials.avatax.rest

**3.2.2. AvaTax Settings Configuration**

Configure Avalara AvaTax related settings using the AvaTax module in the Business Manager.

1. Log into the **Business Manager**.
2. Select the desired site from the dropdown.
3. Navigate to **Merchant Tools > AvaTax > AvaTax Settings**
4. Select/fill all applicable settings for the storefront. Below is the description for these settings in brief.

**AvaTax Options:**

- **Mapped Fields Reporting Tool** – This will open Tax Calculation Field Mappings page where the different object data types are mapped with their associated fields. Configuration and documentation can be changed by referring to sections 4.4. and 4.5 respectively.
- **Tax Calculation** – When enabled, this will enable AvaTax tax calculation on all transactions for the current site.
- **Address Validation** – When enabled, this will enable shipping addresses (both origin and destination) validation during checkout on the storefront. AvaTax only validates addresses in the United States and Canada.
- **Save Transactions to AvaTax** – When enabled, this will save all orders to AvaTax. When disabled, orders will not be saved to AvaTax. However, the tax estimates appear on the storefront.
- **Commit Transactions to AvaTax** - If enabled, successful orders on the storefront will be marked 'Committed' when posting to Avalara AvaTax.

- **Company Code** – This is the AvaTax company code configured on the admin console of the AvaTax account. All successful orders placed on the current storefront will be posted under this company. Configurations such as Importer of Record, Sales Tax nexus of this company will be applicable for all transactions for the current storefront.
- **Customer code** - Select what value is sent as the customer code for authenticated customers. For guest customers, Email ID entered by them during checkout is used.
- **Shipping Tax Code** – If the shipping methods of the storefront do not have tax class assigned, this tax code will be used to calculate taxes for shipping methods. If not specified, AvaTax freight code 'FR' will be used.
- **Validate Address Link** – After filling in the origin address details on Configuration page, the validate address link on being clicked validates the address. After clicking on 'Use Validated Address' button and then clicking on 'Save' button, the address fields of origin address are updated.

### Cross Border Calculation

- **Taxation Policy:** (*Read-only*) If cross border tax/customs duty calculation applies to the storefront, the taxation policy should be Net.

### Address Details

- **Location Code** – If the Location Code value is provided, AvaTax prefers it over other address fields for tax calculation. Location code is configured under Avalara account as company location. If this value is present, the origin address fields can be omitted.
- **Origin (shipFrom) Address fields** – Details of the Company location from where the products will be shipped. These will be used for calculating taxes for all transactions on the current storefront.

Address line 1, line 2, line 3, City, State, Zip/Postal Code, and Country Code should be as per exact Company address to help calculate taxes accurately.

If shipments are being shipped from US or Canada, fill in the complete and valid address details to avoid taxation errors.

#### Note:

- Navigate to **Merchant Tools > Site Preferences > Order > Order Access Settings**, and set **Limit Storefront Order Access** to **No**.
- Make sure to select the required Promotion Preference by Navigating to **Merchant Tools > Site Preferences > Promotion > Discount Taxation**.

### 3.3. Cross Border, VAT, and Invoice Messages

**NOTE:** This section can be skipped if the storefront doesn't ship products from one country to another or does not need to store customs duty or cross border details.

AvaTax service returns taxes and invoice messages based on origin and destination locations in different countries. If the customs duties are applicable, these values are also stored in custom attributes.

## Cross Border Tax Calculation:

The custom duties that are calculated based on origin and destination addresses are stored in custom attributes as follows. These can be utilized in a model to display on the storefront.

Sr. No.	Value	Objects
1.	Custom Duty	<ul style="list-style-type: none"> <li>• basket.custom. ATCustomsDuty</li> <li>• order.custom.ATCustomsDuty</li> </ul>
2.	Taxes	<ul style="list-style-type: none"> <li>• basket.custom.ATTax</li> <li>• order.custom.ATTax</li> </ul>
3.	Cross Border Messages	<ul style="list-style-type: none"> <li>• basket.custom.ATLandedCost</li> <li>• order.custom.ATLandedCost</li> </ul>
4.	Transaction Summary	<ul style="list-style-type: none"> <li>• basket.custom.ATGenericMessage</li> <li>• order.custom. ATGenericMessage</li> </ul>
5.	Invoice Messages	<ul style="list-style-type: none"> <li>• basket.custom.ATInvoiceMessage</li> <li>• order.custom. ATInvoiceMessage</li> </ul>
6.	Tax Break-Up(JSON)	<ul style="list-style-type: none"> <li>• basket.custom.ATTaxDetail</li> <li>• order.custom. ATTaxDetail</li> </ul>

These values are stored as custom attributes in –

1. Basket object, if the order is still not placed
2. Order object, after the current Basket is converted into an Order or the order has been placed

The invoice messages object is stored in below JSON format.

```
{
  "InvoiceMessageMasterList": [
    {
      "MessageCode": 0,
      "Message": "No applicable messaging for this line."
    },
    {
      "MessageCode": 1,
      "Message": "Intra-EU Supply of Goods as per Art. 138 EU VAT Directive 2006/112"
    }
  ],
  "InvoiceMessageList": [
    {
      "TaxLineNo": "98ed5002d09d3b33abba8b10ed",
      "MessageCode": 1
    },
    {
      "TaxLineNo": "f9a79cafa05441b14e72c0e2a6",
      "MessageCode": 0
    }
  ]
}
```

The messages may be utilized to display to the site customer during checkout as required.

Sr. No.	Attribute	Description
1.	InvoiceMessageMasterList	<ul style="list-style-type: none"> <li>It is an array of all applicable messages. 'MessageCode' is the index used to refer to the corresponding 'Message'.</li> </ul>
2.	InvoiceMessageList'	<ul style="list-style-type: none"> <li>It is an array of all line items in the Basket and their corresponding MessageCodes.</li> </ul>
3.	TaxLineNo	<ul style="list-style-type: none"> <li>Indicates UUID of the line items in the current Basket object.</li> </ul>

### For example,

`{"TaxLineNo": "98ed5002d09d3b33abba8b10ed", "MessageCode": 1}` indicates that a line item in the current basket with UUID 98ed5002d09d3b33abba8b10ed has a VAT invoice message 'Intra-EU Supply of Goods as per Art. 138 EU VAT Directive 2006/112' which corresponds to MessageCode 1.

### Tax Breakdown details

When taxes are calculated and returned by Avalara services, it may be displayed to the end user on the storefront.

To accommodate this, AvaTax integration stores the tax details for every line item (Product, Shipping, Gift certificate etc.) in the Basket or Order object in the form of a custom attribute based on whether the order is in placed state or not.

1. If the Basket object is still in checkout process, i.e. If the order is not yet placed, then the tax details are stored in a custom attribute of basket system object.

The attribute name is - `ATTaxDetail`. It can be accessed as - `basket.custom.ATTaxDetail`

2. If the order is in placed state, the tax details are stored on order system object. The attribute name is - `ATTaxDetail`. It can be accessed as - `order.custom.ATTaxDetail`

The tax details are stored in the form of JSON string. It can be utilized as needed. Below is an example of tax details JSON for a Basket.

```
[
  {
    "lineitemid": "75051870
3299",
    "shipmentid": "me",
    "taxes": [
      {
        "jurisdictiontype": "State",
        "jurisdiction": "WASHINGTON",
        "exempt": 0,
        "nontaxable": 0,
        "taxable": 299.9
      }
    ]
  }
]
```

```

        9,
        "rate":0.065,
        "tax":19.5
    },
    {
        "jurisdictiontype":"County",
        "jurisdiction":"KITSAP",
        "exempt":0,
        "nontaxable":0,
        "taxable":299.9
    },
    {
        "jurisdictiontype":"City",
        "jurisdiction":"BAINBRIDGE ISLAND",
        "exempt":0,
        "nontaxable":0,
        "taxable":299.9
    },
    {
        "jurisdictiontype":"State",
        "jurisdiction":"WASHINGTON",
        "exempt":0,
        "nontaxable":0,
        "taxable":9.99,
        "rate":0.065,
        "tax":0.65
    },
    {
        "jurisdictiontype":"County",
        "jurisdiction":"KITSAP",
        "exempt":0,
        "nontaxable":0,
        "taxable":9.99,
        "rate":0,
        "tax":0
    }
]
},
{
    "lineitemid":"me",
    "shipmentid":"me",
    "taxes":[
        {
            "jurisdictiontype":"State",
            "jurisdiction":"WASHINGTON",
            "exempt":0,
            "nontaxable":0,
            "taxable":9.99,
            "rate":0.065,
            "tax":0.65
        },
        {
            "jurisdictiontype":"County",
            "jurisdiction":"KITSAP",
            "exempt":0,
            "nontaxable":0,
            "taxable":9.99,
            "rate":0,
            "tax":0
        }
    ]
}

```

```

    },
    {
      "jurisdictiontype":"City",

      "jurisdiction":"BAINBRIDGE ISLAND",
      "exempt":0,
      "nontaxable":0,
      "taxable":9.99,
      "rate":0.025,
      "tax":0.25
    }
  ]
]

```

### JSON attributes:

Sr. No	Attribute	Description
1.	lineitemid	<ul style="list-style-type: none"> <li>Represents the object ID of the line item from Business Manager. For example, for the product line item, this value would be its Product ID.</li> </ul>
2.	shipmentid	<ul style="list-style-type: none"> <li>Attribute in each object represents the Shipment to which current line item belongs.</li> </ul>
3.	jurisdictiontype	<ul style="list-style-type: none"> <li>Represents the tax jurisdiction of in which the current tax amount is calculated. e.g. Country, State, City, County, Special etc.</li> </ul>
4.	jurisdiction	<ul style="list-style-type: none"> <li>Represents the jurisdiction name. e.g. Canada, Ontario, California etc.</li> </ul>
5.	taxname	<ul style="list-style-type: none"> <li>represents a short tax description for the tax applied. e.g. CANADA GST/TPS, TN STATE TAX etc</li> </ul>
6.	nontaxable	<ul style="list-style-type: none"> <li>has a value which is not taxable taxable attribute indicates the amount that is taxed.</li> </ul>
7.	exempt	<ul style="list-style-type: none"> <li>Attribute indicates the amount which is tax-exempt.</li> </ul>
8.	rate	<ul style="list-style-type: none"> <li>contains the tax rate used for calculating the taxes tax has a value which is the tax that is calculated.</li> </ul>

### 3.4. Services

Some features are also included in this integration as experimental ones. These can be accessed in the AvaTax Settings menu under AvaTax.

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

- **Test Connection** – This can be used to verify whether connection to AvaTax is authenticated/successful. If it indicates that the connection is not authenticated, you may want to have a look at the Service credentials configured in step 3.2.1. On successful test connection, prompt will be shown on configuration page with list of active subscriptions associated to the account.



- **Void Transaction** – This feature can be used void/cancel an order posted earlier to AvaTax. It cancels the order, or it does not, depending on if the order can be cancelled or not. In either case the service response is returned and displayed in the text area below this section.
- **Commit Transaction** – This feature can be used commit an order posted earlier to AvaTax. It marks the order committed or it does not, depending on if the order can be committed or not. In either case the service response is returned and displayed in the text area below this section.
- **Validate Address** – This feature can be used to validate shipping address of an order and receive a validated address as response.

All the above features can be scaled to process many orders on the storefront.

### 3.5. Line Level Addressing

Merchant Seller Identifier can be customized for a specific product as a line item field for customers to track the merchant selling a particular product in a single transaction and can be checked using Retrieve Transaction AvaTax REST v2 API call: `/api/v2/transactions/{id}`.

The merchantSellerIdentifier field has an associated origin address/ship from address, which can be customized using distinct Product ID. This can be customized through a JSON file, from which relevant fields of ship from address and merchantSellerIdentifier will be extracted with respect to products included in a transaction. The JSON file is located at `"\cartridges\int_avatax_svcclient\cartridge\scripts\lineLevelAddress.json"` path.

**JSON attributes:**

Sr. No	Attribute	Description
1.	productID	<ul style="list-style-type: none"> <li>The product ID uniquely identifies each product.</li> </ul>
2.	merchantSellerIdentifier	<ul style="list-style-type: none"> <li>Merchant seller identifier for each product ID.</li> </ul>
3.	shipfrom	<ul style="list-style-type: none"> <li>The shipfrom address for a particular product with address field values (line1, line2, line3, city, region, postalCode, countryCode).</li> </ul>

Using this, customized ShipFrom addresses can be assigned to different products in a single transaction. If a specific product does not have an associated ShipFrom address in JSON file, the origin (corporate) address on Avatax Configuration page is considered as ShipFrom address for that product.

**3.6. Beverages and Alcohol Tax Calculation**

Tax calculation for products under the category of Beverages and alcohol can be done, provided the account has an active AvaAlcohol subscription. This can also be verified by the “test connection successful” prompt which shows the active subscriptions for the current account being used. The user will be able to see AvaAlcohol subscription in the prompt if the user has subscribed to this service.

The attributes related to different BevAlc products need to be customized in the JSON file provided. Distinct products can be assigned parameters which will be extracted during tax calculation if the products are included in a transaction. The JSON file is located at “`cartridges\int_avatax_svcclient\cartridge\scripts\javaBevAlcConf.json`” path.

**JSON attributes:**

Sr. No.	Attribute	Description
1.	productID	<ul style="list-style-type: none"> <li>The product ID uniquely identifies each product.</li> </ul>
2.	parameters	<ul style="list-style-type: none"> <li>Parameters associated to each distinct product ID. The attribute fields included in this are – ContainerSize, AlcoholContent, PackSize, NetVolume, IsAlcoholSample, IsForeignAlcohol and Brand.</li> </ul>
3.	taxCode	<ul style="list-style-type: none"> <li>The Tax Code for a particular product which will determine tax calculation.</li> </ul>

If the ‘RecipientDOB’ is to be updated at product level, the user can fetch it from their own customized cartridges, uncomment the code shown in figure below provided in file: “`cartridges\int_avatax_sfra\cartridge\scripts\helpers\avatax\calculateTaxHelper.js`”, and fill the ‘value’ field of RecipientDOB. The date should be in the format of YYYY-MM-DD.

```
// Uncomment the code below to populate RecipientDOB on product level
// See documentation for more info
// line.parameters.push({
//     "name":"RecipientDOB",
//     "value": ""
// });
```

After tax calculation and successful order placement, these attributes can be verified on the AvaTax CUP portal. The parameters will be reflected on respective product ID to which they have been customized.

### Header level attributes

These attributes are passed at header level. Please uncomment the following block of code present in file:

“cartridges\int\_avatax\_sfra\cartridge\scripts\helpers\avatax\calculateTaxHelper.js” and fill the ‘value’ field. Empty ‘value’ field will generate error.

```
var bevAlcTransactionAttributes = []

// Required attributes for Bev Alc Transaction
// {
//     "name":"AlcoholRouteType",
//     "value": ""
// },

// Required attributes for Returns
// Uncomment only the section you want to populate
// empty "value" will cause error

// {
//     "name":"PurchaserDOB",
//     "value": ""
// },
// {
//     "name":"TrackingNumber",
//     "value": ""
// },
// {
//     "name":"IsDirectAlcoholImport",
//     "value": ""
// },
```

The attributes given in the table below are required for Returns, except 'Varietal'.

Sr. No.	Attribute	Description
1.	AlcoholRouteType	<ul style="list-style-type: none"> <li>This parameter is used to indicate the sales channel for a beverage alcohol transaction.</li> </ul>
2.	PurchaserDOB	<ul style="list-style-type: none"> <li>The date of birth (DOB) associated with the "Bill To" recipient. Must be formatted as YYYY-MM-DD.</li> </ul>
3.	TrackingNumber	<ul style="list-style-type: none"> <li>A text string to indicate the tracking number of transaction.</li> </ul>
4.	IsDirectAlcoholImport	<ul style="list-style-type: none"> <li>Used to indicate whether a beverage alcohol product is a direct import product</li> </ul>
5.	PurchaserName	<ul style="list-style-type: none"> <li>Represents the first and last name of purchaser associated with the transaction</li> </ul>
6.	ShipDate	<ul style="list-style-type: none"> <li>The day merchandise associated with an order was shipped. Must be formatted as YYYY-MM-DD.</li> </ul>
7.	SalesLocation	<ul style="list-style-type: none"> <li>Distinguishes orders that were placed onsite at a business's location vs. offsite</li> </ul>
8.	PurchaserStreetAddress	<ul style="list-style-type: none"> <li>Street address of the purchaser associated with the transaction</li> </ul>
9.	PurchaserToCity	<ul style="list-style-type: none"> <li>City of the purchaser associated with the transaction</li> </ul>
10.	PurchaserToState	<ul style="list-style-type: none"> <li>State of the purchaser associated with the transaction</li> </ul>
11.	PurchaserToZip	<ul style="list-style-type: none"> <li>ZIP of the purchaser associated with the transaction</li> </ul>
12.	CarrierCode	<ul style="list-style-type: none"> <li>Carrier code of shipment</li> </ul>
13.	FullfillmentHouseCode	<ul style="list-style-type: none"> <li>Name of fulfillment house. Required for wine products.</li> </ul>
14.	Varietal	<ul style="list-style-type: none"> <li>For wine only, to indicate what varietal the product is classified</li> </ul>

After tax calculation and successful order placement, these attributes can be verified on the AvaTax CUP portal.

## Hooks

- Hooks have been provided for certain BevAlc functionality, which can be added to customized code where required. These hooks are provided in file: "cartridges\int\_avatax\_sfra\cartridge\scripts\hooks\avatax\avataxhooks.js".
- The 3 hooks added as part of BevAlc orders features are: reverify, cancelOrDeleteShipment and fetchBevAlcConfiguration.
- To use the BevAlc verification features, the user will have to place an initial call using reverify hook on a particular transaction to fetch the verification status.

Sr. No.	Hook	Function/integration
1.	reverify	<ul style="list-style-type: none"> <li>For registering/ overriding BevAlc order.</li> <li>The hook can be added as shown below, with 'orderNo', 'isOverride' (true/false) and 'basket' parameters provided.</li> <li>If the order is compliant, it will be marked as Committed on AvaTax.</li> </ul> <pre data-bbox="505 422 1479 611">if(dw.system.HookMgr.hasHook( "app.custom.bevAlc.shipment.reverify")) {     dw.system.HookMgr.callHook('app.custom.bevAlc.shipment.reverify', 'reverify', {         'orderNo': orderNo,         'isOverride': false,         'basket': basket     }); }</pre>
2.	cancelOrDeleteShipment	<ul style="list-style-type: none"> <li>For cancel/ delete registration of BevAlc order.</li> <li>The hook can be added as shown below, with 'orderNo' and 'deleteShipment' (true/false) parameters provided.</li> </ul> <pre data-bbox="505 793 1479 940">if(dw.system.HookMgr.hasHook( "app.custom.bevAlc.shipment.cancelOrDeleteShipment")) {     dw.system.HookMgr.callHook('app.custom.bevAlc.shipment.cancelOrDeleteShipment', 'cancelOrDeleteShipment', {         'orderNo': orderNo,         'deleteShipment': true     }); }</pre>
3.	fetchBevAlcConfiguration	<ul style="list-style-type: none"> <li>For fetching product level attributes from JSON for a specific productID.</li> <li>The hook can be added as shown below, with 'productID' parameter provided.</li> </ul> <pre data-bbox="505 1087 1479 1161">if(dw.system.HookMgr.hasHook( "app.custom.bevAlc.fetchBevAlcConfiguration")) {     bevAlcAttributes = dw.system.HookMgr.callHook('app.custom.bevAlc.fetchBevAlcConfiguration', 'fetchBevAlcConfiguration', productID); }</pre>

- Sample of how reverify hook can be placed in code:  
If the reverify hook is to be utilized when a particular transaction is placed, the following code modification can be made in file:  
“cartridges\int\_avatax\_sfra\cartridge\scripts\avaTax.js” in function ‘updateTaxDetails’ as shown below:

```
// Persist various values in the objects
var txn = require('dw/system/Transaction');
if (orderNo) {
    if(dw.system.HookMgr.hasHook( "app.custom.bevAlc.shipment.reverify")) {
        dw.system.HookMgr.callHook('app.custom.bevAlc.shipment.reverify', 'reverify', {
            'orderNo': orderNo,
            'isOverride': false,
            'basket': basket
        });
    }
} else if (basket) {
    txn.wrap(
        function () {
            basket.custom.ATInvoiceMessage = invoiceMsgDetails;
        }
    );
}
```

When the reverify hook call is placed, the Verification Status and Verification JSON of the order will be reflected on the Business Manager under **Merchant Tools > Ordering > Orders > Order: Attributes** as shown:

Avalara | AvaTax

Transaction Summary:

Invoice Messages:

Verification Status:

Verification JSON:

Import Duty / Fees:

Tax:

Cross-Border summary:

Tax Break-up (JSON):

- These hooks can be placed where you want to execute the BevAlc functionalities as per the fulfillment process in your code.
- The Verification Status field of the order can have 3 possible values:

Sr. No.	Verification Status	Description
1.	Compliant	<ul style="list-style-type: none"> <li>• This is when the BevAlc order and the parameters passed with the products are compliant with respect to the jurisdiction.</li> </ul>
2.	Non Compliant	<ul style="list-style-type: none"> <li>• This is when the BevAlc order and the parameters passed with the products are non-compliant with respect to the jurisdiction. The customer has the option to re-update the product parameters in the avaBevAlcConf.json file and reverify again.</li> </ul>
3.	Cancelled But Compliant	<ul style="list-style-type: none"> <li>• This is when the BevAlc order and the parameters passed with the products are compliant but the order status has been marked as 'cancelled' on <b>Merchant Tools &gt; Ordering &gt; Orders &gt; Order – General</b>.</li> </ul>

- The Avatax Reconcile Orders page reflects the Verification Status of the BevAlc orders in case of 'Non Compliant' and 'Cancelled But Compliant' orders. The cancel, override and reverify functionality has been provided through the buttons 'Cancel', 'Override' and 'Reverify' on the page, using which the Administrator can perform given functions from Business Manager itself on placed orders. Figure below shows a sample of such orders being populated in the Reconcile Orders table :

Order No.	Order Date	B2C Total Amount	B2C Total Tax	Order Status	AvaTax Total Amount	AvaTax Total Tax	Reconciliation Status	Verification Status
t00001702	2022-12-02	25.99	2.67	CANCELLED	\$25.99	\$2.67	-	Cancelled But Compliant
t00001704	2022-12-02	25.99	2.67	OPEN	N/A	N/A	Missing In AvaTax	Non Compliant

Showing 21 to 22 of 22 entries

Buttons: Cancel, Override, Reverify, Reconcile

### 3.7. External Interfaces

All requests are done through AvaTax REST v2 API. The full reference guide, along with the resource structure for requests, can be found on their developer site - <https://developer.avalara.com/api-reference/AvaTax/rest/v2/>

### 3.8. Firewall Requirements

No firewall changes are required.

### 3.9. Testing

The cartridge also contains unit tests.

You can run `npm test` in the terminal or windows command prompt to execute all unit tests in the project. Unit tests are present under **/test/unit** directory.

## 4.0. Operations, Maintenance

### 4.1. Data Storage

The only Custom Objects stored in Salesforce B2C Commerce are the ones created during the initial import. These are described in the [Configuration section](#).

For data stored by Avalara, please refer to website - <http://www.avalara.com/>

### 4.2. Availability

Cartridge functionality is dependent on the availability of the AvaTax API service. Current AvaTax operational status can be viewed here - <https://status.avalara.com/>

### 4.3. Support

If you encounter any issues related to configuring or testing this integration, you may contact Avalara customer support for assistance. You can submit a case by email, telephone or chat. Please go to <http://avalara.com/Technical-Support> for more details as availability will vary depending on your support package.

### 4.4. Object Mapping

All the mappings related to lines have been put in a single place. To maintain object mapping for calculate tax, please follow the steps below:

1. Go to `cartridges\int_avatax_svcclient\cartridge\scripts\avatax\helpers\calculateTaxHelper.js`
2. Modify the mapping as per the requirement

### 4.5. Object Mapping Documentation

For object mapping documentation, use the following steps:

1. Go to `avataxObjectMapping.html` file under `cartridges\bm_avatax\cartridge\static\default`
2. Find the reference of mapping in the HTML and replace as per the requirements.
3. To create a new mapping, enter a grid-table-row under the table related to the mapping.

```
<div class="grid-table-row">
  <div class="grid-table-cell"> {#} </div>
  <div class="grid-table-cell"> {B2C Commerce Object Data Type} </div>
  <div class="grid-table-cell"> {B2C Commerce Field} </div>
  <div class="grid-table-cell"> {AvaTax Field} </div>
  <div class="grid-table-cell"> {Description} </div>
</div>
```

## 5.0. FAQs

- **How to ensure accurate tax calculation?**
  - Make a note to configure tax codes, UPC codes, and any discounts for products in Business Manager. This information is used by AvaTax to accurately calculate taxes for a product.
  - This can either be manually configured in Business Manager > Merchant Tools > Products module or using a Product feed containing these attribute values.
- **When exactly is tax calculated in the e-commerce journey?**
  - On Shipping page
  - When the end-user fills the shipping address details and continues to the billing page
  - When the end-user changes shipping method (provided that the Basket has shipping address accessible)
- **How many calls are made to AvaTax?**

While the number of calls made to AvaTax are subject to Basket modifications, address modifications, discounts applications etc., the below list indicates a minimum number of calls made to AvaTax during the checkout journey.

- On shipping page, after filling shipping address and continuing to billing page (as SalesOrder document)
  - On Billing page, on filling the billing address (as SalesOrder document)
  - On Placing the order (as SalesInvoice document in uncommitted state)
  - Calls to AvaTax may also be made if the content of the Basket changes in any sense during the checkout process. i.e. If the user adds/removes a product in/from the Basket, or applies a discount, changes an address, or changes the shipping method etc.
- **How to disable AvaTax on a site?**

To disable AvaTax tax calculation and/or address validation for a site –

- Set the values of the Custom Preferences 'Enable AvaTax' and 'Enable AvaTax Address Validation' to No. (**Business Manager > Merchant Tools > Site Preferences > Custom Preferences > AvaTax**)
- Remove any code changes made as a part of AvaTax integration.

- **What happens when the taxes cannot be calculated because of service unavailability, for example?**
  - The tax values are set to Zero instead of fall back to system defined tax rates. This is to ensure that the tax calculation is entirely handled by AvaTax for compliance.

## 6.0. Known Issues

No known existing issues. Please report any issues to [support@avalara.com](mailto:support@avalara.com).

## 7.0. Release History

Version	Date	Changes
1.0	8/8/2010	<ul style="list-style-type: none"> <li>Initial release</li> </ul>
1.1	4/6/2011	<ul style="list-style-type: none"> <li>Shipping Address Validation call added.</li> </ul>
1.2	5/4/2011	<ul style="list-style-type: none"> <li>Shipping Address Validation standalone call added.</li> </ul>
2.0	7/29/2013	<ul style="list-style-type: none"> <li>AvaTax Testing Center added for clients and developers.</li> <li>Test Connection, Commit and Cancel Tax features added.</li> <li>AVS enhancements added.</li> <li>Transaction logging added.</li> <li>Customer and Entity/Use code added.</li> <li>Fixed order level promotion bugs.</li> <li>Fixed documentation and added enhancement steps.</li> <li>Added more custom preferences and configurations for the enhancements displayed above.</li> <li>Cleaned up code and added detailed comments</li> </ul>
2.1	7/16/2015	<ul style="list-style-type: none"> <li>Fixed Multi-Shipping.</li> <li>Implemented use of Web Service Framework Configuration on SFCC Platform.</li> <li>Added metadata files and instructions on importing.</li> </ul>
3.0	8/12/2015	<ul style="list-style-type: none"> <li>Refactored cartridge.</li> </ul>
3.1	6/8/2016	<ul style="list-style-type: none"> <li>Changed Detail Level</li> <li>Updated to use total tax per line instead of recalculating based on tax rate.</li> </ul>
3.2	6/30/2016	<ul style="list-style-type: none"> <li>Added Country Code, Currency Code, and VAT ID (if applicable) to GetTax request.</li> <li>Reduced number of calls made to the service.</li> </ul>
3.3	12/10/2017	<ul style="list-style-type: none"> <li>Added support for controllers.</li> </ul>
19.1	07/12/2018	<ul style="list-style-type: none"> <li>AvaTax REST API support.</li> <li>Added support for VAT and Invoice Messages for EU countries.</li> <li>Reconciliation Utility BM Extension.</li> <li>Commit Transaction on successfully placing order.</li> <li>Custom Customer Code support.</li> </ul>
19.2	06/12/2019	<ul style="list-style-type: none"> <li>Added cross border tax calculation support.</li> <li>New BM tools for AvaTax settings.</li> </ul>

22.1	02/18/2022	<ul style="list-style-type: none"><li>• Origin address validation feature added on Configuration page.</li><li>• Line level addressing for multiple ship from addresses.</li><li>• Logger functionality.</li></ul>
22.2	06/10/2022	<ul style="list-style-type: none"><li>• Beverages and Alcohol tax calculation added.</li><li>• Test connection successful prompt with active subscriptions.</li><li>• Changed lineLevelAddress.json path for line level feature.</li></ul>
22.3	09/08/2022	<ul style="list-style-type: none"><li>• Tax Calculation Code refactoring.</li><li>• Mapped Fields Reporting Tool link added on Configuration page.</li></ul>
22.4	12/09/2022	<ul style="list-style-type: none"><li>• Beverages and Alcohol transaction parameters and returns related functionality added.</li><li>• Beverages and Alcohol transaction related changes on Reconciliation page added.</li></ul>