



Avalara AvaTax for Salesforce B2C Commerce

Headless

User Guide

Revision Date: 05/31/2023

Product Release: 24.1.0

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 - 2024 Avalara, Inc. All rights reserved.

Avalara, AvaTax, Exemption Certificate Management 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. Summary	4
2.0. Component Overview	5
2.1. Functional Overview	5
2.2. Use Cases	5
2.3. Limitations and Constraints	5
2.4. Compatibility	5
2.5. Privacy and Payment	5
3.0. Implementation Guide	6
3.1. Setup	6
3.1.1. Import Cartridges	6
3.1.2. Import Metadata	6
3.1.3. AvaTax Business Manager Extension Cartridge Setup	7
3.2. Configuration	8
3.2.1. Configure AvaTax Services	8
3.2.2. AvaTax Settings Configuration	9
3.2.3. Configure B2C Commerce Headless	11
3.3. Cross Border, VAT, and Invoice Messages	11
3.4. Services	16
3.5. Line Level Addressing	17
3.6. Beverages and Alcohol Tax Calculation	18
3.7. Enhanced VAT	23
3.8. Exemption Utility	25
3.9. Exemption Number	31
3.10. External Interfaces	31
3.11. Firewall Requirements	31
3.12. Testing	31
4.0. Operations, Maintenance	32
4.1. Data Storage	32
4.2. Availability	32
4.3. Support	32
4.4. Object Mapping	32

- 4.5. Object Mapping Documentation.....32
- 5.0. FAQs..... 33
- 6.0. Known Issues..... 35
- 7.0. Release History..... 36

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 Visual Studio Workspace. If using Visual Studio code to import cartridges to the sandbox environment, use the following `dw.json` format file 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",
  "version": "version_to_upload_to"
```

1. Open the Visual Studio Workspace. Browse to the directory where the cartridges have been downloaded.
2. Run the `npm run uploadCartridge` command that would upload ``bm_avatax``, ``int_avatax_sfra``, and ``int_avatax_svcclient`` cartridges to the sandbox you specified in the `dw.json` file. Browse to the directory where the cartridges have been downloaded. (Ensure that the `code-version` or `version` entered in the `dw.json` file matches the version used in the Headless Commerce PWA kit or custom storefront to utilize AvaTax features).
3. Modify the Site Path in **Business Manager > Administration > Manage Sites**. Select your storefront site and go to Settings tab (**Administration > Sites > Manage Sites > Storefront Site –Settings**). Add `'int_avatax_sfra'`, `'bm_avatax'`, and `'int_avatax_svcclient'` cartridges in BM cartridge path.

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

Notes:

- Services named `ceplogger.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 `ceplogger.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:

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

Profile:

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

Service:

Name: *	avatax.rest.all
Type:	HTTP
Enabled:	<input checked="" type="checkbox"/>
Service Mode:	Live
Log Name Prefix:	
Communication Log Enabled:	<input checked="" type="checkbox"/>
Force PRD behavior in non-PRD environments:	<input type="checkbox"/>
Profile:	profile.avatax.rest
Credentials:	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.
- **Enable Certificates Management** - If enabled, exemption certificate utility will be enabled on the storefront for registered customers.
- **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**.

3.2.3. Configure B2C Commerce Headless

- To enable hooks for SCAPI, Navigate to **Administration > Global Preferences > Feature Switches**. Select the 'Enable the execution of hooks in Salesforce Commerce API requests' checkbox and then select Apply.

Feature Switch	Value	Default Value
Enable Case-Insensitive Matching for Suggested Brands	<input checked="" type="checkbox"/> Enable case-insensitive matching for suggested brands.	true
Enable Merged/Individual Display Setting for Variation Groups	<input type="checkbox"/> This setting provides merchandizers the ability to control how search results for Variation Groups are displayed on Storefront search and category navigation.	false
Enable Promotion Refinement	<input type="checkbox"/> Enable this toggle to add search refinements of type Promotion to the product listing pages. Promotions that are marked refinable, and that apply to the products displayed on the product listing page can be added as search refinements.	false
Enable Salesforce Commerce API Hook Execution	<input checked="" type="checkbox"/> Enable the execution of hooks in Salesforce Commerce API requests.	false

- (Skip this step if there is an Headless Commerce instance already running on the sandbox) In order to Configure PWA Kit with B2C Commerce Instance from scratch, refer to the official Salesforce Guide for instructions:

<https://developer.salesforce.com/docs/commerce/pwa-kit-managed-runtime/guide/quick-start.html> .

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"> • basket.custom. ATCustomsDuty • order.custom.ATCustomsDuty
2.	Taxes	<ul style="list-style-type: none"> • basket.custom.ATTax • order.custom.ATTax
3.	Cross Border Messages	<ul style="list-style-type: none"> • basket.custom.ATLandedCost • order.custom.ATLandedCost
4.	Transaction Summary	<ul style="list-style-type: none"> • basket.custom.ATGenericMessage • order.custom. ATGenericMessage
5.	Invoice Messages	<ul style="list-style-type: none"> • basket.custom.ATInvoiceMessage • order.custom. ATInvoiceMessage
6.	Tax Break-Up(JSON)	<ul style="list-style-type: none"> • basket.custom.ATTaxDetail • order.custom. ATTaxDetail

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"> • It is an array of all applicable messages. 'MessageCode' is the index used to refer to the corresponding 'Message'.

Sr. No.	Attribute	Description
2.	InvoiceMessageList'	<ul style="list-style-type: none"> It is an array of all line items in the Basket and their corresponding MessageCodes.
3.	TaxLineNo	<ul style="list-style-type: none"> Indicates UUID of the line items in the current Basket object.

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:

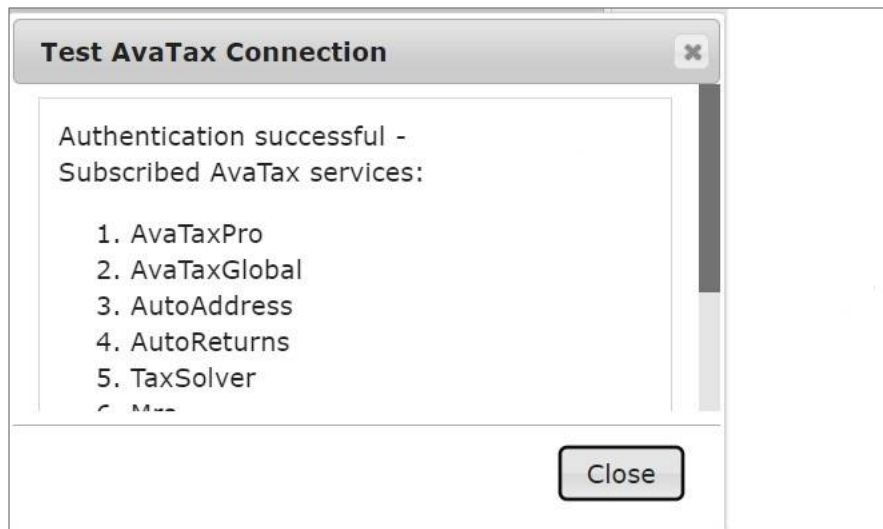
Sr. No	Attribute	Description
1.	lineitemid	<ul style="list-style-type: none"> Represents the object ID of the line item from Business Manager. For example, for the product line item, this value would be its Product ID.
2.	shipmentid	<ul style="list-style-type: none"> Attribute in each object represents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

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"> The product ID uniquely identifies each product.
2.	merchantSellerIdentifier	<ul style="list-style-type: none"> Merchant seller identifier for each product ID.
3.	transport	<ul style="list-style-type: none"> Transport field for each product ID.
4.	shipfrom	<ul style="list-style-type: none"> The shipfrom address for a particular product with address field values (line1, line2, line3, city, region, postalCode, countryCode).
5.	hsCode	<ul style="list-style-type: none"> HS code value for each product ID.

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.

HS Code mapping: HS code can be configured for different product IDs in this JSON file. If the product is added to the cart and the HS code value is populated in the JSON, uncomment the following code in file “`\cartridges\int_avatax_sfra\cartridge\scripts\helpers\avatax\calculateTaxHelper.js`” to enable the passing of the HS code value at the line level:

```
//Uncomment code below to populate hsCode on product level for Cross Border transactions
// See documentation for more info
// if(!empty(jsonAddress.lladdresses[prodID].hsCode)) {
//     line.hsCode = jsonAddress.lladdresses[prodID].hsCode;
// }
```

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\lavaBevAlcConf.json`” path.

JSON attributes:

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

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"> This parameter is used to indicate the sales channel for a beverage alcohol transaction.
2.	PurchaserDOB	<ul style="list-style-type: none"> The date of birth (DOB) associated with the "Bill To" recipient. Must be formatted as YYYY-MM-DD.
3.	TrackingNumber	<ul style="list-style-type: none"> A text string to indicate the tracking number of transaction.
4.	IsDirectAlcoholImport	<ul style="list-style-type: none"> Used to indicate whether a beverage alcohol product is a direct import product
5.	PurchaserName	<ul style="list-style-type: none"> Represents the first and last name of purchaser associated with the transaction
6.	ShipDate	<ul style="list-style-type: none"> The day merchandise associated with an order was shipped. Must be formatted as YYYY-MM-DD.
7.	SalesLocation	<ul style="list-style-type: none"> Distinguishes orders that were placed onsite at a business's location vs. offsite
8.	PurchaserStreetAddress	<ul style="list-style-type: none"> Street address of the purchaser associated with the transaction
9.	PurchaserToCity	<ul style="list-style-type: none"> City of the purchaser associated with the transaction
10.	PurchaserToState	<ul style="list-style-type: none"> State of the purchaser associated with the transaction
11.	PurchaserToZip	<ul style="list-style-type: none"> ZIP of the purchaser associated with the transaction
12.	CarrierCode	<ul style="list-style-type: none"> Carrier code of shipment
13.	FullfillmentHouseCode	<ul style="list-style-type: none"> Name of fulfillment house. Required for wine products.
14.	Varietal	<ul style="list-style-type: none"> For wine only, to indicate what varietal the product is classified

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"> For registering/ overriding BevAlc order. The hook can be added as shown below, with 'orderNo', 'isOverride' (true/false) and 'basket' parameters provided. If the order is compliant, it will be marked as Committed on AvaTax. <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"> For cancel/ delete registration of BevAlc order. The hook can be added as shown below, with 'orderNo' and 'deleteShipment' (true/false) parameters provided. <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"> For fetching product level attributes from JSON for a specific productID. The hook can be added as shown below, with 'productID' parameter provided. <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: [Text Area]

Invoice Messages: [Text Area]

Verification Status: **Compliant**

Verification JSON: [{"compliant": true, "warningCodes": ["UnsupportedTaxCode"], "r

Import Duty / Fees: **0**

Tax: **2.67**

Cross-Border summary: [Text Area]

Tax Break-up (JSON): [{"lineitemid": "wine-12", "shipmentid": "me", "taxes": [{"jurisdictiontype": "State", "jurisdiction": "WASHINGTON", "exempt": 0, "nontaxable": 0, "taxable": 10, "rate": 0.065, "tax": 0.65, "taxName": "WA STATE TAX", "taxSubType": "U", "taxType": "Use"}, {"jurisdictiontype": "County", "jurisdiction": "KING", "exempt": 0, "nontaxable": 0, "taxa

- 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"> • This is when the BevAlc order and the parameters passed with the products are compliant with respect to the jurisdiction.
2.	Non Compliant	<ul style="list-style-type: none"> • 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.
3.	Cancelled But Compliant	<ul style="list-style-type: none"> • 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 Merchant Tools > Ordering > Orders > Order – General.

- 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:

Showing 10 entries

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. Enhanced VAT

The configurations and fields listed below have been added as part of Enhanced VAT for Marketplaces to record the necessary information to AvaTax during transactions.

- Is Seller Importer Of Record:** When you successfully import the metadata during the configuration of cartridges, the parameter **isSellerImporterOfRecord** is a checkbox available in Business Manager's customer profile page. Go to **Merchant Tools** → **Customers** → **Customers** and select a customer profile. The **Is Seller importer of record** option is available under the AvaTax section. If you select it, then the value is set to True, otherwise it remains False. The default value is set to False. These details are fetched when a registered customer with that specific customer number places a transaction on the storefront. It is also recorded at the transaction level.

The AvaTax section appears as follows:

- Business Identification Number:** The 'businessIdentificationNo' value can be set in Business Manager by the administrator, using the Tax ID section on the Customer Profile page (the same page as shown above for isSellerImporterOfRecord).
- Merchant-related fields:** The Merchant Seller Identifier field has associated merchant parameters, which you can customize using the distinct merchantSellerIdentifier values. You can customize it using a JSON file, from which relevant fields of merchant-related parameters (MerchantEstablishmentCountry, MerchantName and MerchantVatId) are extracted with respect to the merchantSellerIdentifier values included in a transaction. The JSON file is located at the following path:
 “\cartridges\int_avatax_svcclient\cartridge\scripts\avaMerchantConf.json”.

- JSON Attributes:**

Sr. No.	Attribute	Description
1.	merchantSellerIdentifier	<ul style="list-style-type: none"> Merchant seller identifier for each product ID.
2.	MerchantEstablishmentCountry	<ul style="list-style-type: none"> Merchant Establishment Country for a specific merchantSellerIdentifier.
3.	MerchantName	<ul style="list-style-type: none"> Merchant Name for a specific merchantSellerIdentifier.

Sr. No.	Attribute	Description
4.	MerchantVatId	<ul style="list-style-type: none"> Merchant Vat ID for a specific merchantSellerIdentifier.

- Marketplace Liability Type and Origination Site:** The attributes 'marketplaceLiabilityType' and 'originationSite' are passed at the transaction level. Uncomment the following block of code in the following figure, present in the file: "cartridges\int_avatax_sfra\cartridge\scripts\helpers\avatax\calculateTaxHelper.js" and fill the value. Accepted values for marketplace liability type are Seller and Marketplace.

```

line.taxOverride = null;

// Uncomment the code below to populate marketplaceLiabilityType and originationSite on product level
// See documentation for more info
// line.marketplaceLiabilityType = "";
// line.originationSite = "";

//Check if customer is bev-alc registered
if(verifyBevAlcEnabled()){

```

- Point Of Order Origin Address:** The pointOfOrderOrigin field is passed at the product line level in a transaction. Its value is that of the billing address entered by a customer during the checkout process.
- Transport:** The Transport parameter value can be configured at the transaction (header) level or the product (line) level or both. To configure it on the transaction level, uncomment the following block of code in the file "cartridges\int_avatax_sfra\cartridge\scripts\avaTax.js" under function 'calculateTax' and add the transport value as follows:

```

// Uncomment the code below to populate Transport field in Transaction Parameters on Header
// See documentation for more info
transactionModel.parameters.push({
  "name": "Transport",
  "value": "Seller"
});

// get the hash for transactionModel
var hash = murmurhash.hashBytes(JSON.stringify(transactionModel), JSON.stringify(transactionModel).length, 523);

```

The parameter is sent to AvaTax when tax calculation is triggered. Additionally, the transport value can be customized on the product level using a JSON file. The relevant fields of transport and merchantSellerIdentifier are extracted from the JSON file with respect to the products included in the transaction. The JSON file is located at the following path: "cartridges\int_avatax_svcclient\cartridge\scripts\lineLevelAddress.json".

(This is also mentioned under Section 3.5 in the attributes table). The Transport field on the header and line level or both can be customized and used at the same time.

The transport field can have only these possible values: 'Seller', 'Buyer', 'ThirdPartyforSeller', 'ThirdPartyforBuyer' or 'None'. Any invalid or empty value passed will generate an error.

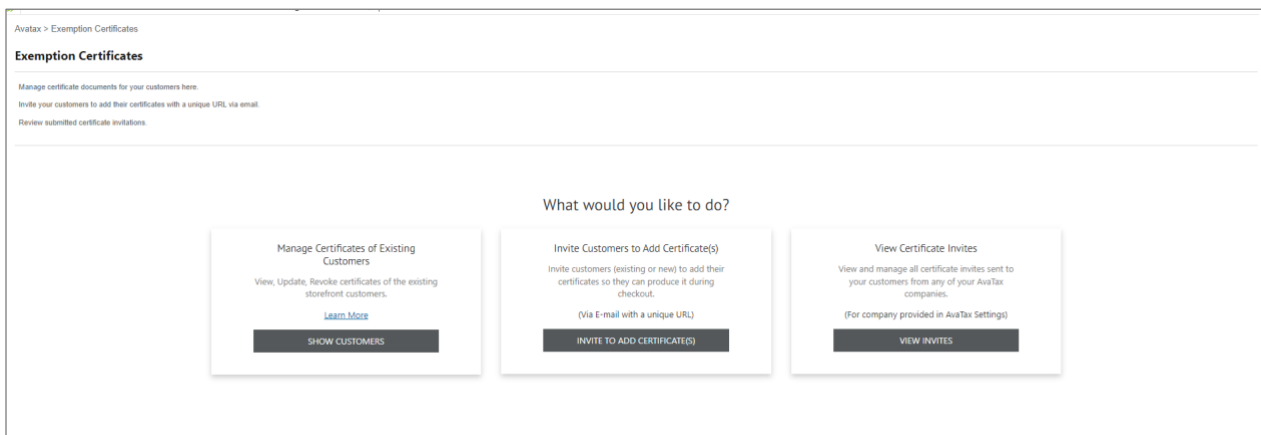
- **Invoice messages:** Any invoice message provided by AvaTax in response to the transaction placed, is populated in the Orders Summary page on Business Manager in the custom Avalara 'Attributes' section. This has also been explained previously in the Section 3.3.
- **VAT Code:** VAT code can be retrieved for a particular transaction using the response received from AvaTax. The object 'svcResponse' in the file "cartridges\int_avatax_sfra\cartridge\scripts\avaTax.js" under the function 'processTaxDetails' contains the details of the transaction and can be used to populate the response data by customers wherever required according to their business use case.
- **Tax Included:** The tax inclusive feature and the backend storefront logic of the site depends on the taxation policy of the site being used in Salesforce B2C (Net site or Gross site) and they affect the response. Taxation policy of a particular site can be viewed on Business Manager by going to **Merchant Tools** → **Ordering** → **Taxation** or on the AvaTax Settings page under Taxation Policy (Read-only). If the Taxation policy is 'Gross', the site is tax inclusive.

The 'taxIncluded' value is populated from the B2C site settings and is passed at all the line-level items of a transaction, in the file "cartridges\int_avatax_sfra\cartridge\scripts\helpers\avatax\calculateTaxHelper.js". If the site is tax inclusive, the value is set to 'true', else it is 'false'.

3.8. Exemption Utility

Business Manager Exemption Utility

Go to **Merchant Tools > AvaTax > Exemption Certificates**. The following Exemption Certificates page is displayed, provided the account has an active AvaCert subscription. This can be verified by the "test connection successful" prompt which shows the active subscriptions for the current account that is being used. You can see the AvaCert subscription in the prompt if the user has subscribed to this service.

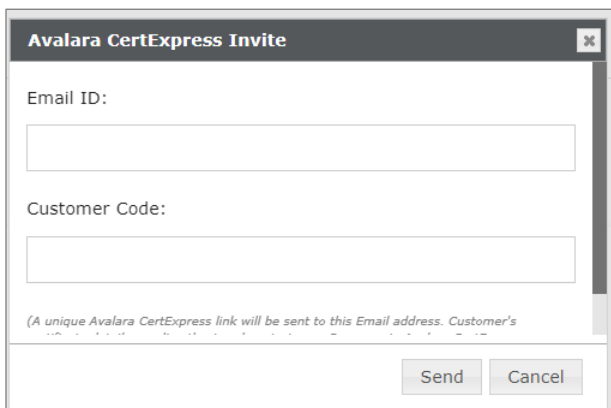


From the Exemption Certificates landing page, you have the following three options:

- Manage Certificates of Existing Customers by selecting the **Show Customers** button.

This redirects you to a page with a table consisting of the existing storefront customers and their account details - Customer Number, Name, Email, and Customer Since. To view a particular customer, view the certificates attached to them, and the related functions, select any customer number in the table. It will redirect to the Customer page. More information on this page is available under **Customer Details Page** point below.

- Invite Customers to Add Certificate(s) through an E-mail with a unique URL by clicking the **Invite to Add Certificate(s)** button. This opens a popup box where the admin can enter the Email ID and Customer Code of the customer, they wish to send the email to and select **Send**. An Avalara CertExpress link is sent to this email address which they can use to add a certificate.



The screenshot shows a modal window titled "Avalara CertExpress Invite". It contains two text input fields: "Email ID:" and "Customer Code:". Below the fields is a small line of text: "(A unique Avalara CertExpress link will be sent to this Email address. Customer's". At the bottom of the modal are two buttons: "Send" and "Cancel".

- View a list with details of Certificate Invites associated with the current active company being used by clicking the **View Invites** button. It redirects you to the Certificate Invites page where it displays the list of invite details as a table.

Customer Details Page

After you select a particular customer number on the customers list table, it redirects you to the Customer details page (AvaTax > Exemption Certificates > Customer).

If the specific customer is available on the storefront but the record is not available in AvaTax, it displays the **Create Customer Record** button on the page. Selecting this button opens a pop-up in which you can enter customer details and then select the **Create** button on the pop-up, to create the customer record in AvaTax. The details provided in this pop-up affects only AvaTax values and it does not modify the storefront details of the customer.

Create Customer Record

(The details below are retrieved from the customer record in Salesforce.)

Name:*

Email ID:*

Address Line 1:*

Address Line 2:

City:*

Postal Code:*

Region/State:*

On successful creation of customer record, the page refreshes and displays the certificate action buttons. If there are any certificates already attached to that customer number, it will be populated as a table, else the page appears as follows if no certificates data for the customer is found:

Avalara > Exemption Certificates > Customer

Exemption Certificates

This module shows all certificates linked to the customer.

Click [Invite to Add Certificate\(s\)](#) button to invite your customer to add their certificates via an email with a unique link.

Select the [Update Customer Details](#) option to update customer profile information in AvaTax.

Click [Create Certificate\(s\)](#) to create a certificate record for each customer in Avalara AvaTax.

Select a row to take action on one record at a time.

Hover over buttons to know more about what they do.

ID	Name	Actions
123456	user1, test	Invite to Add Certificate(s) Update Customer Details Create Certificate(s)

No certificates data found for this customer.

The admin can now use certificate related utility provided:

- **Invite Customer:** On clicking the Invite to Add Certificate(s) button, a pop-up appears to provide the email address of the customer.

On adding the Email address value and **selecting** Send, an email containing CertExpress link **is sent** to that customer requesting to add a new certificate.

- **Update Customer Details:** Select the **Update Customer Details** button to update the customer record in AvaTax. A pop-up opens with the current customer values on AvaTax auto populated which the admin can edit. Upon entering the new values and selecting the **Update** button on the pop-up, the AvaTax customer details are updated, and a success message is notified on the button.

- **Create Certificate:** By selecting the **Create Certificate(s)** button, a pop-up appears with a dropdown list of exposure zones. Selecting a specific zone initializes GenCert on the pop-up. It asks for details valid to that exposure zone and create the certificate for the customer with final success message shown. To know more about the GenCert feature and its utilities, view its documentation. The pop-up will be

shown as below when GenCert initializes, and tax information of the customer is displayed. During the certificate creation process, if you want to change the exposure zone at any time, use the exposure zone dropdown list to select another country. On selecting another country, GenCert re-initializes according to the details asked for the new exposure zone.

Create Certificate

Record an exemption certificate document for this customer.

Select a new state/exposure zone

Exposure Zone:*

Alabama

(Note: When you create a certificate, it will be processed by Avalara and will become available for use in calculating tax exemptions when processing is complete.)

Your Tax Information

NAME
test user1

PHONE

ADDRESS
247 Soscol Ave

CITY

Close

Upon successful certificate creation, a success message appears on the pop-up with a Print/Download link which downloads the certificate created by GenCert. You can select **Close** and on refreshing the page, the table will contain a certificate record for that exposure zone.

Avalara > Exemption Certificates > Customer

Exemption Certificates

This module shows all certificates linked to the customer.
 Click Invite to Add Certificate(s) button to invite your customer to add their certificates via an email with a unique link.
 Select the Update Customer Details option to update customer profile information in AvaTax.
 Click Create Certificate(s) to create a certificate record for each customer in Avalara AvaTax.
 Select a row to take action on one record at a time.
 Hover over buttons to know more about what they do.

123456 / user1, test

Invite to Add Certificate(s) Update Customer Details Create Certificate(s)

Show 10 entries Search

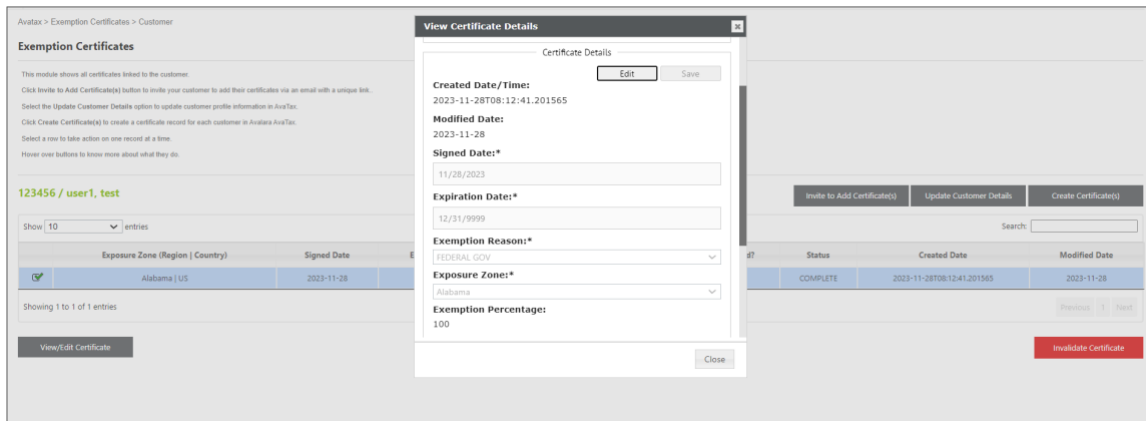
	Exposure Zone (Region Country)	Signed Date	Expiration Date	Document Exists?	Valid?	Verified?	Status	Created Date	Modified Date
<input type="checkbox"/>	Alabama US	2023-11-28	9999-12-31	Yes	Yes	No	COMPLETE	2023-11-28T08:12:41.201565	2023-11-28

Showing 1 to 1 of 1 entries Previous Next

View/Edit Certificate Print/Download Certificate

- View/Edit Certificate Details:** The **View/Edit Certificate** button is displayed on the page when at least a certificate record is available for the customer. On selecting one certificate record at a time and selecting the **View/Edit Certificate** button, a pop-up containing certificate details populated is shown. To edit the fields, select the

Edit button as shown below on the pop-up and the details which can be updated for the certificate will become editable. When the new values have been entered, select the **Save** button on the pop-up. This saves the details and shows a success message under the **Close** button. Upon selecting **Close**, the page will refresh if record had been successfully updated and changes will be visible on the table.



- Invalidate Certificate:** The **Invalidate certificate** button is displayed on the page when at least one certificate record is available for the customer. On selecting a certificate record at a time and selecting the button, a popup asking for confirmation to proceed with invalidation appears. By selecting **Proceed**, the certificate gets invalidated, and a success message is displayed under the **Invalidate** button. On refreshing the page, the certificate record is no longer listed in the table.

GenCert Configuration

GenCert provides some optional parameters that can be set according to admin preferences. Further information on how they can be utilized is provided in GenCert documentation:

https://knowledge.avalara.com/bundle/sgv1667483188945_sgv1667483188945/page/Install_AvaCert_for_eCommerce.html.

To configure these parameters for Create certificate feature in:

- Business Manager:** Under **Avatax > Exemption Certificates > Customer**, for the Create Certificate feature, GenCert parameters can be configured in the file 'cartridges\bm_avatax\cartridge\static\default\js\avcustomerdetail.js'. Some optional parameters have been added, for example, to enable the option to print/download a certificate upon successful creation by setting the attribute 'show files' to true. If this attribute is set to false, the **Print/Download certificate** link that appears under success message upon certificate creation will not be visible.

```

/** Initialising GenCert with required parameters. For optional
GenCert.init(
    document.getElementById("form_container"),
    {
        token: createcerttoken,
        ship_zone: selectedCountry,

        /** Optional GenCert parameters. */

        debug: false,
        show_cert_ids: true,
        show_files: true,
        preview: false,
    }
);
GenCert.show();
});

```

3.9. Exemption Number

The **Exemption Number** field has been provided on the Customer profile page in Business Manager. Upon successful import of metadata provided in the AvaTax B2C package, the exemption number field will be added to the customer profile page and visible on the UI. Go to **Merchant Tools > Customers > Manage Customers** and select a customer. On the customer profile page, under the AvaTax section, there will be a field **Exemption Number** which the admin can populate. This field will be sent during checkout for the registered customer and if the value is populated, the tax will be exempt, and details can be viewed on the CUP portal for verification.

The screenshot shows a form with the following elements:

- AvaTax** section:
 - Is seller importer of record:
 - Exemption Number:
- Customer Preferences** section:
 - Preferred Locale: (dropdown menu)
 - Note:

3.10. 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/>

3.11. Firewall Requirements

No firewall changes are required.

3.12. 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.
- **What happens when an order fails during checkout after taxes have been calculated?**
 - If an order fails at any point of the checkout process due to some unexpected error (payment failure, internet issues etc.), the order details are not captured on the AvaTax Transactions page and error logs are placed. Only after successful checkout, the sales invoice can be placed in AvaTax.

6.0. Known Issues

No known existing issues. Please report any issues to support@avalara.com.

7.0. Release History

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

19.2	06/12/2019	<ul style="list-style-type: none"> Added cross border tax calculation support. New BM tools for AvaTax settings.
22.1	02/18/2022	<ul style="list-style-type: none"> Origin address validation feature added on Configuration page. Line level addressing for multiple ship from addresses. Logger functionality.
22.2	06/10/2022	<ul style="list-style-type: none"> Beverages and Alcohol tax calculation added. Test connection successful prompt with active subscriptions. Changed lineLevelAddress.json path for line level feature.
22.3	09/08/2022	<ul style="list-style-type: none"> Tax Calculation Code refactoring. Mapped Fields Reporting Tool link added on Configuration page.
22.4	12/09/2022	<ul style="list-style-type: none"> Beverages and Alcohol transaction parameters and returns related functionality added. Beverages and Alcohol transaction related changes on Reconciliation page added.
23.1	03/17/2023	<ul style="list-style-type: none"> Fixed invoice logic to place sales invoice of Transaction to AvaTax only upon successful order checkout.
23.2	09/15/2023	<ul style="list-style-type: none"> Added Enhanced VAT functionality.
23.3	12/01/2023	<ul style="list-style-type: none"> Exemption utility for Business Manager and Storefront Exemption number field utility HS code field mapping for cross border transactions
24.1	05/31/2024	<ul style="list-style-type: none"> First compatible build for Salesforce B2C Commerce Headless