



# Avalara AvaTax for Microsoft Dynamics 365 for Finance and Operations

---

## System Integration Document for Avalara AvaTax

Version 1.1

Revision date: 12/13/18

*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.*

© 2017 Avalara, Inc. All rights reserved.

*Avalara, AvaTax, AvaTax Calc, AvaTax Returns, AvaTax Certs, AvaTax Local, AvaLocal, AvaTax POS, AvaPOS, AvaRates, TrustFile, BPObridge, Laserbridge+, Sales TaxII, Sales TaxPC, SalestaxPC+, StatetaxII, and StatetaxPC 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>AvaTax Integration Points</b> .....	<b>3</b>
Integrations on Address Validation on Masters .....	3
Integrations on Transactions (Module - Accounts receivable/Sales & Marketing) .....	3
Integrations on Transactions (Module - Project management & accounting) .....	3
Integrations on Transactions (Module – Accounts Payable).....	4
Address Validation for SOAP .....	5
Address Validation Example for SOAP .....	5
Address Validation for REST .....	6
Address Validation Example for REST.....	6
Calculate Tax for Transaction .....	7
Tax Calculation Example .....	7
Avalara Avatax tax Service field mapping for sales order .....	8
Delegates in REST .....	10
Delegates Example .....	10

## AvaTax Integration Points

Avalara AvaTax has been implemented in the following area which helps you to understand the customized area for AvaTax.

### Integrations on Address Validation on Masters

Page Name	Address Validation	Latitude / Longitude
Legal entities	Yes	Yes
Customer Master	Yes	Yes
Vendor Master	Yes	Yes
Sites Master	Yes	Yes
Warehouses Master	Yes	Yes

**Note:** We had integration on LogisticalPostalAddress form, which is a base form for all address UI.

### Integrations on Transactions (Module - Accounts receivable/Sales & Marketing)

Form Name	Address validation	Latitude / Longitude	Entity/ Use Code Mapping	Tax Override	Sales Tax
Quotation List Page	Yes	Yes	Yes		Yes
Quotation	Yes	Yes	Yes		Yes
Sales Order List Page	Yes	Yes	Yes		Yes
Sales Order	Yes	Yes	Yes		Yes
Sales Order List Page (With order type as Return Order)	Yes	Yes	Yes	Yes	Yes
Sales Order (With order type as Return Order)	Yes	Yes	Yes	Yes	Yes
Free Text invoices List Page	Yes	Yes	Yes		Yes
Free Text Invoice	Yes	Yes	Yes		Yes

### Integrations on Transactions (Module - Project management & accounting)

Form Name	Address validation	Latitude / Longitude	Entity/ Use Code Mapping	Sales Tax
Project Sales Order List Page	Yes	Yes	Yes	Yes
Project Sales Order	Yes	Yes	Yes	Yes
Project Invoice proposal - Journals - Fee, Expense, Hours, Item - Sales order - Service Order				Yes

**Integrations on Transactions (Module – Accounts Payable)**

Form Name	Address validation	Latitude / Longitude	Use Tax Assessment
Purchase Order List page			
Purchase Order			
Vendor Invoices List Page			
Vendor Invoice			Yes

## Address Validation for SOAP

If you want to use Avalara Address validation:

1. Create an object of AVA\_AddressValidation class.
2. Call the validateAddress(str\_line1, str\_line2, str\_line3, str\_city, str\_state, str\_zip, str\_country).
  - line1, line2, line3 are the lines of Street field of the address.
3. To get the validated address, create an object of 'Addressing' (Addressing is standard EDT of AX) and assign the parmAddressing() of AVA\_AddressValidation.
  - addressingobject = AVA\_AddressValidation.parmAddressing();

## Address Validation Example for SOAP

The following example displays a validated address using Avalara Address Validation service.

```
static void addressValidationDemo(Args _args)
{
    Addressing                validAddress;
    LogisticsPostalAddress    postalAddress;

    AVA_AddressValidation    addressValidDemo = new AVA_AddressValidation();

    postalAddress = LogisticsPostalAddress::findRecId(52565474129); //Address which need to
    be validated.

    addressValidDemo.validateAddress(strLine(postalAddress.Street,0),
                                    strLine(postalAddress.Street,1),
                                    strLine(postalAddress.Street,2),
                                    postalAddress.City,
                                    postalAddress.State,
                                    postalAddress.ZipCode,
                                    postalAddress.CountryRegionId);

    validAddress = addressValidDemo.parmAddressing(); //Validated Address.

    info(strFmt("%1",validAddress));
}
/** Example InfoLog output
Message (03:34:04 pm)
900 Winslow Way E
Bainbridge Island,WA 98110-2450
USA
*/
```

## Address Validation for REST

If you want to use Avalara Address validation:

1. Create an object of AVAAddressValidation class.
2. Call the validateAddress(LogisticsPostalAddress).
  - LogisticsPostalAddress Table
3. To get the validated address, create object of 'Addressing' (Addressing is standard EDT of AX) and assign the parmAddressing() of AVAAddressValidation.
  - addressingobject = AVAAddressValidation.parmAddressing();

## Address Validation Example for REST

The following example displays a validated address using Avalara Address Validation service.

```
static void addressValidationDemo(Args _args)
{
    Addressing                validAddress;
    LogisticsPostalAddress    postalAddress;

    AVAAddressValidation      addressValidDemo = new AVAAddressValidation();

    postalAddress = LogisticsPostalAddress::findRecId(52565474129); //Address which need to
    be validated.

    addressValidDemo.validateAddress(postalAddress);

    validAddress = addressValidDemo.parmAddressing(); //Validated Address.

    info(strFmt("%1",validAddress));
}
/** Example InfoLog output
    Message (03:34:04 pm)
    900 Winslow Way E
    Bainbridge Island,WA 98110-2450
    USA
    */
```

## Calculate Tax for Transaction

If you want to calculate tax through Avalara tax service

1. Create an object of Table – SalesTable (you can create an object of the different table as per below transaction type), Ava\_Header. Class - AVA\_TaxEngine, a Real variable to store the tax value.

Table	Transaction type
salesTable	Sales Order
custInvoiceTable	Free text invoice
salesQuotationTable	Sales quotation
projProposalJour	Project invoice proposal

2. Initialize the table record with the order number.
3. Call the GetTax (Common\_table, TransDate\_documentDate= DateNull ())
  - ‘\_documentdate’ is date for which you want to calculate the sales tax.
  - ‘\_table’ can have different value based on the document for which you want to calculate the tax.
4. Fetch the ‘TaxAmount’ field value from the AVA\_Header table and assign it to the tax variable.

## Tax Calculation Example

The following example displays a validated address using Avalara Address Validation service.

```

static void taxCalcDemo(Args _args)
{
    SalesTable      salesTable;
    real            salesTaxAmount;
    AVA_Header      headerTable;

    AVA_TaxEngine   taxEngine = new AVA_TaxEngine();

    salesTable      = salesTable::find('000882'); //Sales order for which tax need to be
    calculated.

    taxEngine.GetTax(salesTable, systemDateGet());

    select taxAmount from headerTable
        where headerTable.DocNumber == '000882'; //Fetching the tax value.

    salesTaxAmount = headerTable.TaxAmount; //Assigning the tax value.

    info(strFmt("Sales Tax : $ %1",salesTaxAmount));
}
/** Example InfoLog output
Message (01:35:56 pm)
Sales Tax : $ 45.27
*/

```

## Avalara Avatax tax Service field mapping for sales order

Header Information	
AvaTax Field Name	Sales Order
CompanyCode	configuration.UserDefinedCompanyCode?configuration.UserDefinedCompanyCode:current();
DocType	SalesOrder
DocCode	salesTable.SalesId
DocDate	SystemDateGet()
SalespersonCode	hcmWorker::find(salesTable.ReclId).PersonnelNumber
CustomerCode	salesTable.CustAccount
Discount	Total discount of the document
PurchaseOrderNo	salesTable.PurchOrderFormNum
ExemptionNo	salesTable.VATNum
Origin Address	There is a fallback mechanism first system will look for header Warehouse address >> header Site id Address >> Current AX Company address
Destination Address	salesTable.custTable_CustAccount().postalAddress()
DetailLevel	
LocationCode	inventLocation::find(InventDim::find(salesLine::find(salesTable.SalesId,1).InventDimId).InventLocationId).InventLocationId
Commit	FALSE
CurrencyCode	salesTable.CurrencyCode
TaxOverride	Tax override type, reason, amount and tax override true



<b>Line Information</b>	
<b>AvaTax Field Name</b>	<b>Sales Order</b>
No	Unique line number for each line
Origin address	There is a fallback mechanism first system will look for Line Warehouse address >> Line Site id Address >> Current AX Company address
Destination address	salesLine.deliveryAddress()
ItemCode	SalesLine.ItemId
TaxCode	Tax code from AVA_TaxCodeMapping table with fall back mechanism Item >> Item group >> SalesCategory >> Item sales tax group
Qty	salesQuantity::construct(documentStatus::Invoice).calcSalesQty(salesLine,specQty)
Amount	salesLine.calclineAmountExclTax(qty, Systemdateget())
RevAcct	DimensionStorage::ledgerDimension2AccountNum(InventPosting::itemLedgerDimension(InventAccountType::SalesRevenue,"",InventItemGroupItem::findByItemIdLegalEntity(salesLine.ItemId,salesLine.dataArealD).ItemGroupId))
ExemptionNo	salesTable.VATNum
CustomerUsageType	ava_EntityUseCodeMapping::find(salesLine.CustAccount, LogisticsPostalAddress::findReclD(salesLine.DeliveryPostalAddress).Location).EntityTypeCode
Description	salesLine.Name
BusinessIdentificationNo	taxRegistration.RegistrationNumber assign to salesLine.deliveryAddress()

## Delegates in REST

If you want to customize the tax calculation parameter, then you can use a delegate.

We have created `fetchHeaderData_delegate` and `fetchLineData_delegate` for the following class

- AVATaxEngine\_CustInvoice
- AVATaxEngine\_Payable
- AVATaxEngine\_Project
- AVATaxEngine\_ProjectSO
- AVATaxEngine\_Receivable
- AVATaxEngine\_SalesQuotation

How to use a delegate to update the origin/ship from the address of the header?

1. Create an extension class.
2. Copy `fetchHeaderData_delegate` method event handler method of a delegate from above mention classes.
3. Create an object of `AddressValidationInfo`.
4. Get the address using the `parmOriginAddress` method.
5. Update address.

## Delegates Example

The following example displays how to use delegates to update the header origin address.

```
[ExtensionOf(classStr(AVATaxEngine_Receivable))]
final class Delegate_Extension
{
    /// <summary>
    ///
    /// </summary>
    /// <param name="_avaTaxEngine_Receivable"></param>
    /// <param name="_result"></param>
    [SubscribesTo(classStr(AVATaxEngine_Receivable),
    delegateStr(AVATaxEngine_Receivable, fetchHeaderData_delegate))]
    public static void
    AVATaxEngine_Receivable_fetchHeaderData_delegate(AVATaxEngine_Receivable
    _avaTaxEngine_Receivable, EventHandlerResult _result)
    {
        Avalara.AvaTax.RestClient.AddressValidationInfo    addressfromLoc;

        addressfromLoc = _avaTaxEngine_Receivable.parmOriginAddress();
        addressfromLoc.line1 = '1597 W Prospector Way ';
        addressfromLoc.city = ' Queen Creek ';
        addressfromLoc.region = ' AZ';
        addressfromLoc.postalCode = '85242-4455';
        addressfromLoc.country = 'US';
    }
}
```