



AvaTax SDK for IBM i

Implementation Guide

Version 1.0.1

Revision date: 06/07/2013

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.

© 2013 Avalara, Inc. All rights reserved.



Avalara, AvaTax Calc, AvaTax Returns, AvaTax Certs, 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.

Contents

Introduction.....	4
The Business Case	4
Document Scope	4
Who Should Read this Document	5
Features and Benefits of the AvaTax SDK for IBM i	5
What Does the AvaTax SDK for IBM i Do?	5
Prerequisites	6
Requirements and Constraints	6
Software and Hardware Requirements.....	6
General Constraints	10
AvaTax Admin Console	11
AvaTax Admin Console and Web Service.....	12
Web Service Access	13
Client Profile Information	14
Installing the AvaTax SDK for IBM i	15
Install the AVATAX Library onto your IBM Server.....	15
For pre-V6R1	16
Authorizing Users to the AvaTax SDK for IBM i Menu System.....	16
Configure the System Control File	19
GetTax Example	20
Testing the Connection to Avalara Web Services	23
To Test the Connection to Avalara Web Services	23
SDK Integration Tester	26
Integration Tester Defaults	27
Developing Your Connector	29
Web Service Operations.....	29
Ping	30
GetTax	30
PostTax / CommitTax	32
CancelTax.....	32
AdjustTax	32
Interfacing with AvaTax SDK for IBM i.....	32
HttpI - HTTP Parameters Used by the Web Service Call	33

Asial – Order/Invoice Parameters.....	33
AsiaO – Tax Data Returned by AvaTax.....	33
TxJuO – Tax Jurisdiction Data Returned by AvaTax.....	33
Required Fields.....	34
Common System Messages.....	35
Coding Considerations.....	37
Other Considerations.....	38
Tax Compliance.....	50
Nexus Jurisdictions.....	51
ItemCode versus TaxCode.....	52
Pre - GoLive Check List.....	53
Documents.....	53
Item Codes and Description.....	53
Nexus Jurisdictions.....	53
Server Audit.....	53
Appendix A: HTTP Parameters.....	54
Appendix B: Order/Invoice Parameters.....	55
Appendix C: Tax Data Parameters.....	60
Appendix D: Tax Jurisdiction Data Parameters.....	65
Avalara Support.....	66

Introduction

Avalara provides automated sales tax solutions to streamline cumbersome, error-prone tax compliance processes and reduce the risk of loss or penalty in case of an audit. Our solutions automatically perform address validation, jurisdiction research, and rate calculation and allow you to manage even the most complicated tax issues, such as situs¹, nexus, tax tiers, tax holidays, exemptions, certificate management, and product taxability rules.

AvaTax Calc is a powerful, online sales tax compliance solution. The AvaTax SDK for IBM i or SDK adapter allows you to easily create integration with AvaTax Calc. It is designed to operate with virtually any AS400, iSeries or System i application where tax calculations or address validation is required.

The SDK adapter makes the AvaTax Calc, AvaRates, and Address Validation web services available to customers and developer partners alike, providing full testing and support services as well as optional time-saving connector examples.

The Business Case

Accurately calculating sales and use tax reports for the myriad of jurisdictions where companies conduct business is complicated, time consuming, and confusing. Large corporations with hefty specialized tax or accounting departments use expensive, complex software systems to compute rates and file required reports. But medium-sized and smaller businesses can rarely afford the investment such "enterprise" systems require. End users are forced to manually find, collect, and update information, and if they do it incorrectly, may be exposed to serious consequences of non-compliance.

Avalara is committed to freeing companies from their manual and error-prone processes today with the costs associated with adding new clients, time consuming tax rate research, and tax schedule maintenance. The AvaTax solution delivers best practices in sales and use tax compliance, provides companies with improved internal controls, and is designed to integrate into existing order management systems providing the best end-user experience. Avalara provides you with a rich tax API in the SDK adapter to achieve the best possible end-user experience. The SDK adapter is fully documented, including sample code and complete certification requirements.

Document Scope

This document is restricted to describing implementation of the SDK adapter product and the requirements of the System i environment needed for the SDK adapter to consume Avalara web services.

¹ In law, the situs (pronounced "sī'tēs") (Latin for position or site) of property is where the property is treated as being located.

Who Should Read this Document

Any developer with an interest connecting to Avalara web services using the SDK adapter should read this document. They will create the '*blue*' ERP-specific connector shown in Figure 1.1 below. The '*green*' AvaTax SDK for IBM i provided by Avalara handles building the XML SOAP request, sending it to the service, and parsing the returned XML response.

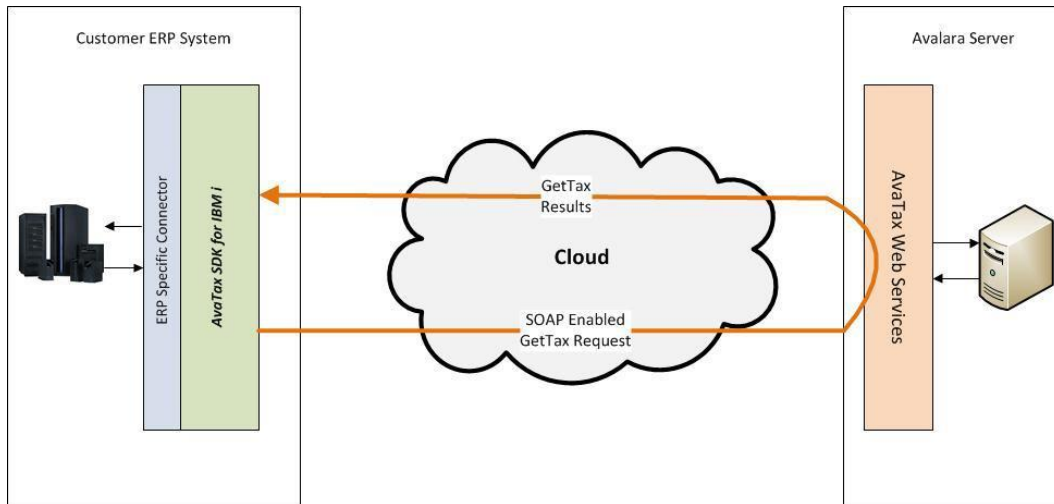


Figure 1.1

Features and Benefits of the AvaTax SDK for IBM i

The SDK adapter:

- is written entirely in RPG. There is no need for developers to use or understand Java. Further, there is no need to implement IBM Websphere.
- uses W3C-recommended Simple Object Access Protocol (SOAP) 1.1 and WSDL best practices.
- runs on all versions of the System i operating system from V5R3.
- has a small footprint. It requires the installation of a single library of about 150MB.
- handles all Internet communications with Avalara web services.
- is supported by Avalara.

What Does the AvaTax SDK for IBM i Do?

A typical use would be as follows:

1. An application passes an invoice to the SDK adapter for which a tax calculation is required.
2. The SDK adapter receives the invoice details and generates a SOAP XML request which it transmits over the Internet to the Avalara 'GetTax' web service.
3. The SDK adapter then receives the response message from the service, parses it, and returns the calculated tax results back to the calling application. As a call is made to the AvaTax Calc web service, details of the transaction can be recorded in the customer's cloud-based AvaTax Admin Console for later reconciliation and/or tax filing purposes.

Prerequisites

Several prerequisites must be in place before using the AvaTax SDK for IBM i.

Requirements and Constraints

Following are requirements and constraints associated with the SDK adapter. Check that these are satisfied before beginning installation.

Software and Hardware Requirements

- A System i (AS/400, iSeries, or i5) running OS/400 version V5R3 or later.
- TCP/IP installed and configured. To check TCP/IP installation, type GO LICPGM at a command line and look for the entry highlighted in Figure 2.0.

For example, 5761TC1 IBM TCP/IP Connectivity Utilities for i5/OS

For more information, see:

<http://pic.dhe.ibm.com/infocenter/iseries/v6r1m0/index.jsp?topic=/rzaij/rzaijlantcp.htm>

```

Display Installed Licensed Programs
System: TAX1

Licensed Program   Installed Status   Description
5761LSV            *COMPATIBLE        Extended Integrated Server Support
5761NAE            *COMPATIBLE        Network Authentication Enablement
5761PT1            *COMPATIBLE        IBM Performance Tools for i5/OS
5761PT1            *COMPATIBLE        Performance Tools - Manager Feature
5761PT1            *COMPATIBLE        Performance Tools - Agent Feature
5761PT1            *COMPATIBLE        Performance Tools - Job Watcher
5761QU1            *COMPATIBLE        IBM Query for i5/OS
5733SC1            *INSTALLED         IBM Portable Utilities for i5/OS
5733SC1            *INSTALLED         OpenSSH, OpenSSL, zlib
5761ST1            *COMPATIBLE        DB2 Query Mgr and SQL DevKit
5761TC1            *COMPATIBLE        IBM TCP/IP Connectivity Utilities for i5/OS
5761TS1            *COMPATIBLE        IBM Transform Services for i5/OS
5761TS1            *COMPATIBLE        Transforms - AFP to PDF Transform
5770UME            *COMPATIBLE        5770UME 00

More...

Press Enter to continue.

F3=Exit  F11=Display release  F12=Cancel  F19=Display trademarks

```

Figure 2.0

- You must ensure that all users will have access to the key database which is part of the certificate store and is stored in the IFS.

If you are using the *SYSTEM certificate store, which is the normal case, then the database can be found in the /QIBM/USERDATA/ICSS/CERT/SERVER directory of the IFS. If you're using a different certificate store, then it will be located wherever you decided to put it.

In either case you will need to make sure any AvaTax users have access to the key databases. The easiest way to do this is to grant *PUBLIC access to the directories and key databases. Users will need read (*R) access to the key database, and read/execute (*RX) access to each of the nested directories that it is in.

Use the WRKLNK command to access each object link and use option 9 to work with authorities.

e.g. WRKLNK 'QIBM/USERDATA'

This displays the following. Enter option 9 to display the 'Work with Authority' screen.

```

Work with Object Links

Directory . . . . : /qibm

Type options, press Enter.
  2=Edit  3=Copy  4=Remove  5=Display  7=Rename  8=Display attributes
 11=Change current directory ...

Opt  Object link      Type      Attribute  Text
 9   UserData         DIR

Parameters or command
===>

F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position to
F22=Display entire field  F23=More options
Bottom

```

```

Work with Authority

Object . . . . . : /qibm/UserData
Type . . . . . : DIR
Owner . . . . . : QSYS
Primary group . . . . . : *NONE
Authorization list . . . . . : *NONE

Type options, press Enter.
  1=Add user  2=Change user authority  4=Remove user

Opt  User          Data Authority  --Object Authorities--
      |            |          | Exist  Mgt  Alter  Ref
-----|-----|-----|-----|-----|-----|
-     *PUBLIC      *RX
-     QSYS         *RWX      X      X      X      X
-     QDIRSRV      *X
-     QTIVOLI      *RX

Bottom

Parameters or command
==>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve
F11=Display detail data authorities  F12=Cancel  F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 2007.

```


So QIBM, USERDATA, ICSS, CERT and SERVER will need to have authority that looks like Fig.2.1a below. User *PUBLIC *RX.

```

Work with Authority

Object . . . . . : /qibm/UserData/ICSS/Cert/Server
Type . . . . . : DIR
Owner . . . . . : QSYS
Primary group . . . . . : *NONE
Authorization list . . . . . : *NONE

Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

Opt  User      Data      --Object Authorities--
    User      Authority  Exist  Mgt  Alter  Ref
  --
  -- *PUBLIC  *RX
  -- QSYS    *RWX      X      X      X      X

Bottom

Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve
F11=Display detail data authorities   F12=Cancel   F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 2007.

```

Fig. 2.1a

The 2 'DEFAULT' files in the Server directory will need to have the public authority shown in Fig. 2.1b *PUBLIC *R

```

Work with Authority

Object . . . . . : /qibm/UserData/ICSS/Cert/Server/DEFAULT. >
Type . . . . . : STMF
Owner . . . . . : QSYS
Primary group . . . . . : *NONE
Authorization list . . . . . : *NONE

Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

      Data      --Object Authorities--
Opt  User      Authority  Exist  Mgt  Alter  Ref
--
-    *PUBLIC    *R
-    QSYS       *RW        X      X      X      X
-

Parameters or command
==>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve
F11=Display detail data authorities  F12=Cancel   F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 2007.
  
```

Fig.2.1b

General Constraints

Two versions of the AvaTax SDK for IBM i are available. Download and install the one that is appropriate for your environment.

Version 1

To allow compatibility with V5R4 of OS/400 the following represents the limits for a single tax call:

- 100 detail lines
- 15 addresses

NOTE

- *These are not the absolute limits for a single invoice or order since they can be overcome by programming techniques within the 'blue' ERP-specific connector.*

Version 2

Version 2 of the AvaTax SDK for IBM i can be installed on operating systems at V6R1M0 or greater. The limits for a single tax call are increased compared to Version 1 as follows:

- 999 detail lines
- 99 addresses

AvaTax Admin Console

Tax profiles as well as other company organizational settings are managed within the AvaTax Admin Console (a secure SSL website). Tax reporting is also a function of the AvaTax Admin Console. You will need to have an AvaTax Admin Console account to use Avalara web services. Credentials and instructions are provided to you as part of your order fulfillment. You can sign up for a free trial account by going to Avalara.com.

Training

It is a recommended best practice that individuals using the AvaTax Admin Console - as well as the developers of SDK custom connectors - attend a free [Avalara University](https://Avalara.com) webinar. Webinars cover the AvaTax Admin Console functions as well as most major ERPs.

Classes are held regularly and appear for sign up at the [Avalara University](https://Avalara.com) site. Please select the class that is appropriate for your company's subscription (Pro or Basic) and the ERP used.

Developers, although not necessarily maintaining company settings in the AvaTax Admin Console, are highly encouraged to take these same webinars to gain a thorough understanding of the AvaTax Admin Console and how it relates to the results and actions of the SDK custom connectors.

AvaTax Admin Console and Web Service

To Login to the AvaTax Admin Console

1. Enter the **URL** to the **Development** AvaTax Admin Console: <https://admin-development.avalara.net/>

NOTE

- The **Production** AvaTax Admin Console URL is <https://admin-avatax.avalara.net/>

2. Enter your **User Name** (for example, avalara.sample@avalara.com)
3. Enter the **Password** that was sent to you via email

NOTES

- This password is temporary. You are prompted to enter a new password.
- If you have lost this password, click the *Forgot your Password?* link on the login screen.

The screenshot shows the AvaTax Admin Console login page. At the top left is the AvaTax logo. Below it is the heading "Log in." followed by the text "Welcome to AvaTax. Log in to manage your account." There is a section for "New Users" with instructions to retrieve user name and password. Below this are input fields for "*User Name:" and "*Password:", a "Forgot your Password?" link, and a "LOG IN" button. To the right of the login form is a yellow box titled "1099 Reporting. Managed in the cloud." with a "GET STARTED" link. Below that is a grey box titled "Sales Tax Secrets: How to Tax Craisins" with a "READ MORE..." link. At the bottom left is the Avalara logo and copyright information. At the bottom right is a "LOGIN LINKS" section with links to "Customer Center", "AvaTax Certs", and "AvaTax 1099".

Figure 2.1

Web Service Access

All information needed to connect and authenticate to the Avalara web services on the development server is contained in the system control file (AX0001P). You just need to add the Account and Password provided by Avalara. When you decide to go live with your application you need to make sure you have entered your production credentials and changed the End Point URL to point to the Avalara production server (<https://admin-avatax.avalara.net/>).

```

AC001R1                      Edit System Control File Record                      4/16/13
                                                                           12:14:23
Company: 4 APITrialCompany          Id: 5 001          Service Active: Y
Debug: N Debug File Name: _____ (e.g. /tmp/mydebug.txt)
      Port: 443 Acct: 1 Your account #          Pass: 2 .....
End Point: 3 https://development.avalara.net/Tax/TaxSvc.asmx
Service Provider: 6 Avalara Inc.          ERP Ver: 7 Infor XA v6.1
Adapter Ver: 8 V1.0          Client System: 9 IBM System i
HTTP User Headers and Data:          No. Hdrs: 1
1: SOAPAction
   "http://avatax.avalara.com/services/GetTax"
2: _____
3: _____
Remove IFS Files: N Address Validation: N Address Conversion Type: Upper
F3=Exit  F12=Cancel
MA  a          MW          03/011
  
```

Figure 2.2

- Account Number

This is your account number that needs to set in the credentials portion of your connector. You need to set this in your system control file in order to use it (for example, 1100012345). See Figure 2.2, number **1**.

- Password

This needs to set in the password field of your system control file. See Figure 2.2, number **2**.

- URL

https://development.avalara.net is the URL of the **development account service**. This must be called out in your connector program but in order to make things simpler the correct development URL is provided in your system control file. See Figure 2.2, number 3.

https://avatax.avalara.net is the URL of the **production account service**. When you are ready to go live, you can re-direct to this URL by simply changing the End Point in your system control file.

- AvaTax WSDL

The AvaTax WSDL is viewable at <https://development.avalara.net/Tax/TaxSvc.wsdl>

Client Profile Information

The Client Profile is a required element in your AvaTax SDK integrations for several good reasons. You should set these details up after installing your AvaTax library at the same time as you enter your company ID, account number and password.

The values passed in the client profile fields:

1. Aid in troubleshooting connector problems
2. Identify OEM transactions for developer / partner programs
3. Provide the developer with a method to version their adapters and locate them in an enterprise environment

The profile properties should be set up in your system control file as follows:

- Service Provider

Set to 'Avalara Inc.'. See Figure 2.2, number 6.

- ERP Version

Set to your ERP and version number. See Figure 2.2, number 7.

- Adapter version

Maintained by Avalara. See Figure 2.2, number 8.

- Client System

Set to indicate the type of IBM server you use (for example, AS400). See Figure 2.2, number 9.

Installing the AvaTax SDK for IBM i

This section explains how to install SDK adapter to your system, setup your system control file, and test the service connection.

Install the AVATAX Library onto your IBM Server

Components for SDK adapter are contained in a single library called AVATAX. The first step is to install the AVATAX library onto your system. AVATAX is made available as an IBM save file which can be downloaded from the Avalara website and installed onto your system using these instructions.

NOTE

- *There are two versions of the AVATAX library available. It is very important that you download the correct one for your IBM operating system. Regardless of which one you use they should both be restored to your server as AVATAX.*
- For V6R1M0 and above download AVATAX61.
- For pre-V6R1M0 systems download AVATAX.

To load the save file onto your System i machine:

1. Sign on to your System i with command line access.
2. Create an empty AVATAX61 save file in the library you would like to download to (for this example we use library QGPL, so use the command CRTSAVF QGPL/AVATAX61).
3. Download your save file using FTP. At a command line type the following:
 - a. FTP 'ftp.taxrates.com'
 - b. User: ftp_avatax
 - c. Password: Avalara!
 - d. BIN
 - e. CD FTP500_Avatax
 - f. LCD QGPL
 - g. GET AVATAX61 (replace
 - h. QUIT
4. RSTLIB SAVLIB(AVATAX61) DEV(*SAVF) SAVF(QGPL/AVATAX61) RSTLIB(AVATAX)

For pre-V6R1

1. Sign on to your System i with command line access.
2. Create an empty AVATAX save file in the library you would like to download to (for this example we use library QGPL, so use the command CRTSAVF QGPL/AVATAX).
3. Download your save file using FTP. At a command line type the following:
 - a. FTP 'ftp.taxrates.com'
 - b. User: ftp_avatax
 - c. Password: Avalara!
 - d. BIN
 - e. CD FTP500_Avatax
 - f. LCD QGPL
 - g. GET AVATAX (replace
 - h. QUIT
4. RSTLIB SAVLIB(AVATAX) DEV(*SAVF) SAVF(QGPL/AVATAX)

Authorizing Users to the AvaTax SDK for IBM i Menu System

The AvaTax SDK for IBM i has its own menu system which allows easy access to the maintenance and inquiry functions. The first time you try access the menu and add new users you will need to sign on with the QSECOFR user profile.

Menu Security gives control over which users have access to the SDK adapter menu items.

Setting up menu security **must** be performed in order to use the SDK adapter application.

Files:

- SC0005P – Table of User Profiles/Authorized Menu Options
- SC0010P – Menu Options

Program:

- SC0005R – User Authority Maintenance

Command:

- ASIA – start the menu system

To start the menu security program:

1. Add AVATAX to your library list (ADDLIB AVATAX *LAST).
2. Type 'ASIA' at the command line and press <Enter>.
3. This brings up the AvaTax SDK for IBM i menu shown in Figure 3.1 below.

```
MAIN                               AvaTax SDK for IBM i                               4/16/13
GWILSON                           Main Menu                               11:58:08

1. Maintenance Menu
2. Inquiry Menu
3. Reporting Menu

11. Administration Menu

20. Integration Tester Defaults
21. Integration Tester

23. Work with Spool Files
24. Signoff

Enter Option: __

F3=Exit
Copyright 2013 Avalara, Inc. All Rights Reserved

MA a                               MW                               19/016
```

Figure 3.1

4. Select option 11 – Administration Menu

At the next screen take option 2 – Menu security. The screen shown in Figure 3.2 below is displayed. From here you can change, copy, delete, display, and add users.

NOTE

- Remember to press F5 to update!

```
SC0005-A          AvaTax SDK for IBM i          4/16/13
GWILSON           User Authority Maintenance     12:08:03

Type Option, Press Enter.
2=Change   3=Copy   4=Delete   5=Display
Opt  User      User Text
-    GWILSON    Graham Wilson user profile
-    QSECOFR    Security Officer
-    RONB       Ron Brown

Bottom

F3=Exit      F6=Add
Copyright 2013 Avalara, Inc. All Rights Reserved
MA  a          MW          09/003
```

Figure 3.2

Configure the System Control File

The AvaTax SDK for IBM i System Control file contains all the details for performing HTTPS/SSL communications with Avalara web services. It also allows you to control some behavioral features of the application. For example, you can turn address validation on or off, or switch the system to work in debug mode.

The file is delivered to you with most of the key data already populated. However, you need to add the elements specific to your environment, such as your account number and password.

To do this, bring up the AvaTax SDK for IBM i menu system as shown in Figure 3.1 Select option **1. Maintenance Menu** and then option **1. Maintain System Control File** from the Maintenance Menu that appears. This displays the screen shown in Figure 3.3 below.

```

AC001R1                Maintain System Control File                4/16/13
                                                                12:12:03

Type option, press Enter.
 2=Edit  3=Copy  4=Delete  5=Display  T=Test

Opt Primary Service      Service Description      Company Code
- Tax Services          GetTax              APITrialCompany
- Tax Services          PostTax             APITrialCompany
- Tax Services          CommitTax           APITrialCompany
- Tax Services          CancelTax           APITrialCompany
- Tax Services          AdjustTax           APITrialCompany
- Tax Services          Ping               APITrialCompany
- Address Services      Validate            APITrialCompany
- Address Services      Ping               APITrialCompany

F1=Info  F3=Exit  F6=Add
Copyright 2004-2013 Avalara, Inc. All Rights Reserved
MA  a                               MW                                07/003

```

Figure 3.3

This screen shows all the Avalara web services available to you. These are discussed in detail later on but the most important ones you will use are GetTax and Validate. Regardless of which services you decide to use, you need to edit each of them as described in the following example for GetTax, before you use the system.

GetTax Example

Put a 2 next to the service you wish to edit and press <Enter>. This displays a screen like Figure 3.4.

```
AC001R1                      Edit System Control File Record                      4/16/13
                                                                    12:14:23
Company: APITrialCompany          Id: 001          Service Active: Y
Debug: N Debug File Name:                  (e.g. /tmp/mydebug.txt)

Port: 443 Acct: Your account #          Pass: .....
End Point:
https://development.avalara.net/Tax/TaxSvc.asmx

Service Provider: Avalara Inc.          ERP Ver: Infor XA v6.1
Adapter Ver: V1.0          Client System: IBM System i

HTTP User Headers and Data:          No. Hdrs: 1
1: SOAPAction
   "http://avatax.avalara.com/services/GetTax"
2:                 
3:                 

Remove IFS Files: N Address Validation: N Address Conversion Type: Upper

F3=Exit F12=Cancel

MA a MW 03/011
```

Figure 3.4

The following is an explanation of the fields you can change.

- **Company**

This is provided with a default of APITrialCompany and you will almost certainly need to change it. This must match the company code that you will be using from your ERP which must also match the Company that you set up in your AvaTax Admin Console (see Figure 3.5).

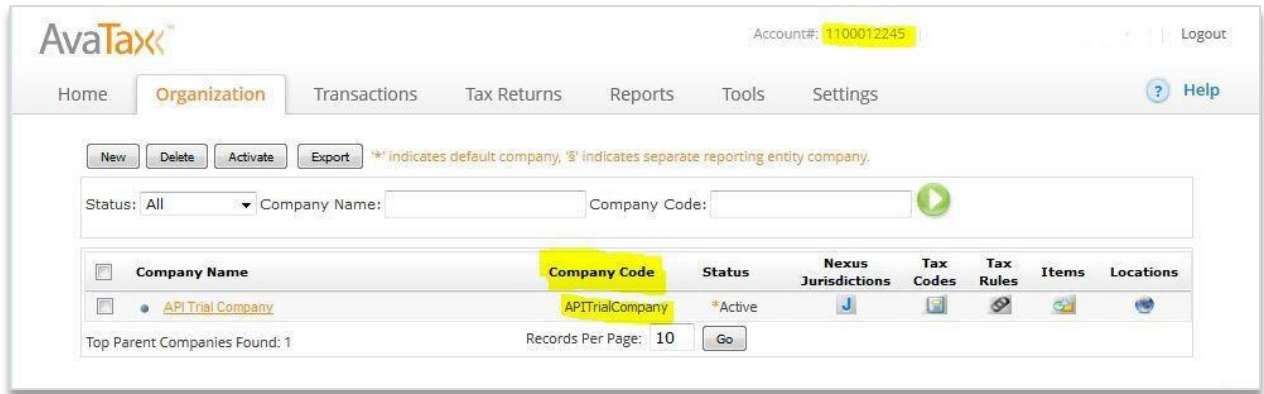


Figure 3.5

- **Id**

This is the service ID assigned by Avalara. You cannot change this field.

- **Service Active**

This switch allows you quickly and easily turn off the service. You can enter Y for active or N for disabled.

- **Debug**

This allows you to switch the SDK adapter into debug mode. When set to Y, a debug report is placed on your Integrated File Server (IFS) in the file specified in the **Debug File Name**.

Debug File Name

If left blank the debug file name will default to `/tmp/httpapi_debug.txt` which is the text file `httpapi_debug.txt` in folder `/tmp` on the IFS. Otherwise specify your own folder and file name.

NOTE

- *The folder name you specify must already exist on the IFS.*

- **Port**

This is the TCP port used for HTTPS communications with the Avalara web service. Port 443 is the default port for use with SSL and you should leave this untouched.

- **Acct**

This should be set to the account number that was emailed to you (see Figure 3.6)

- **Pass**

Set this to the license key that was emailed to you (see Figure 3.6).

NOTE

- *This field is non-display. The easiest way to enter it is to copy and paste from the email.*

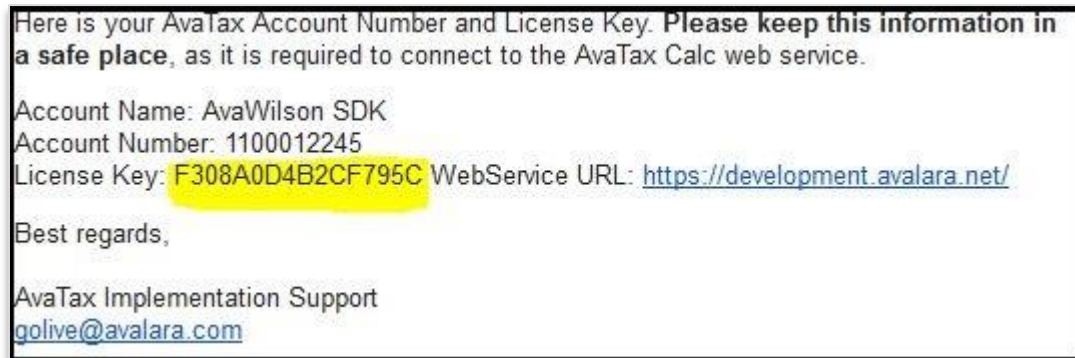


Figure 3.6

- **End Point**

This is the end point for the service. By default it is set to our development server. You will only need to change this when you go live in production.

Development URL: <https://development.avalara.net>

Production URL: <https://avatax.avalara.net>

- **Service Provider**

This is set to 'Avalara Inc.' and can be left alone.

- **ERP Ver**

Set this to a description of your ERP and version.

- **Adapter Ver**

This is informational only and is maintained by Avalara.

- **Client System**

Set this to a description of your IBM server.

- **HTTP User Headers and Data**

These are the SOAPAction details. Do not change unless instructed by Avalara.

- **Remove IFS Files**

Reserved for future use. Leave as N.

- **Address Validation**

Set to Y if you have subscribed to our address validation service and wish to validate addresses before making a GetTax call. Otherwise set to N.

- **Address Conversion Type**

As well as performing address validation, the address validation service can return a sanitized address in different formatting. Choose either **Default**, **Upper**, or **Mixed**. Leaving this field blank will result in the **Default** method being used.

Testing the Connection to Avalara Web Services

Now that you have set up the system control details, you are ready to test your connection to the service.

To Test the Connection to Avalara Web Services

1. From the Maintain System Control File main screen put a 'T' next to the service you want to test and press <Enter>.

```

AC001R1                               Maintain System Control File                               4/16/13
                                                                                               12:28:05

Type option, press Enter.
  2=Edit  3=Copy  4=Delete  5=Display  T=Test

Opt Primary Service      Service Description      Company Code
  T Tax Services         GetTax                  APITrialCompany
  _ Tax Services         PostTax                 APITrialCompany
  _ Tax Services         CommitTax               APITrialCompany
  _ Tax Services         CancelTax               APITrialCompany
  _ Tax Services         AdjustTax               APITrialCompany
  _ Tax Services         Ping                   APITrialCompany
  _ Address Services     Validate                APITrialCompany
  _ Address Services     Ping                   APITrialCompany

F1=Info  F3=Exit  F6=Add
Copyright 2004-2013 Avalara, Inc. All Rights Reserved

MR  a                               MW                               08/003

```

2. If you have correctly configured your service you will get a 'Success' response as shown in Figure 3.7.

```
AC001R1                               Maintain System Control File                               4/16/13
                                                                                               12:28:05

Type option, press Enter.....
 2=Edit 3=Copy 4=Delet:      Test Connection Settings Result      :
                             :                                     :
Opt Primary Service      :      Result: Success                  :
 I Tax Services          :                                     :ny
- Tax Services          :      Tran. ID: 350606580          :ny
- Tax Services          :                                     :ny
- Tax Services          :      Version: 13.2.0.16          :ny
- Tax Services          :                                     :ny
- Tax Services          :      F12=Cancel                  :ny
- Address Services      :                                     :ny
- Address Services      :.....:ny

F1=Info  F3=Exit  F6=Add
Copyright 2004-2013 Avalara, Inc. All Rights Reserved

MA  a                               MW                               08/003
```

Figure 3.7

3. If you get the message shown in Figure 3.8 then re-check your **Acct:** and **Pass:** details.



Figure 3.8

If you are having trouble getting a successful connection then please contact our support team for assistance. See [Avalara Support](#) at the end of this document

However, assuming you received the '**Success**' message then have completed installation and configuration, you ready to start developing your connector.

SDK Integration Tester

Once your connection to the service has tested OK, you can use the SDK Integration Tester to generate transactions on your AvaTax Admin Console account for the purposes of testing, learning more about how the system works, or working out a business scenario that you may wish to implement.

The tester is available as from the first screen of the AvaTax SDK for IBM i menu. It looks like the example in Figure 3.9 and allows you to create most of the transaction scenarios allowed by the AvaTax web service.

```

AX006R1                      SDK Integration Tester                      4/16/13
Dft: 01                      12:37:39

Ship TO:                      Ship FROM:
Line1: 1S278 Wenmoth          100 Ravine Lane
Line2:
City: Batavia                Bainbridge Island
State: IL Zip: 60510          WA 98110

Service: 001 DocId:          <- Orig ->  <-- New -->
Co: APITrialCompany DocTyp: SalesInvoice DocCde:
DocDate: 2013-04-16 CustNo: TEST01 Dis Amt: Exemp No:
EntUse: _ PO#: P00000001 SalPerCd: 123 Locn: Ref: Commit: N

ORTyp: None Rsn: Amt: Date: 2013-04-16
CancelCode: PostAmt: PostTax:
AdjustRsn: 0 AdjustDesc:

# Item Code Tax Inc Dsc? Use Code Qty Amount Rate% Tax Total
1 Boat N N - 1 1000.00
2 Car N N - 2 5000.00
3 Train N N - 1 8000.00
-----

F1=Help F3=Exit F4=Refresh
MA a MW 02/008

```

Figure 3.9

Integration Tester Defaults

For your convenience we have provided an option to allow you to save your favorite test scenarios. This can save you the need for repetitive typing when testing.

```
AX007R1                Maintain Test Defaults - Select                4/16/13
                                                                    12:38:37

Type option, press Enter.
 2=Edit  3=Copy  4=Delete

Opt  Key  Description
--  --  --
  _   01  SalesInvoice
  _   02  SalesOrder
  _   03  SalesOrder - Tennessee

F1=Help  F3=Exit  F6=Add
MA      a                MW                                07/003
```

Select, edit, copy, delete, or add default templates.

```

AX007R1          SDK Integration Tester Defaults          4/16/13
                                                         12:40:25

Dft: 01  Description: SalesInvoice
      Ship TO:                               Ship FROM:
Line1: 1S278 Wenmoth                          100 Ravine Lane
Line2:                                     _____
      City: Batavia                          Bainbridge Island
      State: IL  Zip: 60510                     WA  98110

Service: 001  Co: APITrialCompany DocTyp: SalesInvoice  Commit: N

CustNo: TEST01  PO#: P0000001  SalPerCd: 123

#  Item Code    Tax    Ent  Tax
#  Item Code    Inc Dsc? Use Code      Qty  Amount
1  Boat         N     N   -   -         1  1000.00
2  Car          N     N   -   -         2  5000.00
3  Train        N     N   -   -         1  8000.00

F1=Help  F3=Exit  F6=Add
MA  a                               MW                                03/008

```

Populate as required and save with a unique Dft: value (for example, 01 above).

To use the template type the Dft: code into the first field of the Integration Tester (Figure 3.9) and press <Enter>. The screen is populated with the appropriate data.

Developing Your Connector

This section gives guidance and examples of how to use the SDK adapter to connect to Avalara web services. If you remember from Figure 1.1, the SDK adapter handles the building of the SOAP request, performs the HTTP transmission, and parses the returned XML returning the results to your program in an easily manageable data structure.

The more complex parts of planning your integration with the SDK adapter will be determining:

- When to call the AvaTax web service.
- When to update your ERP's internal tables with the results data that are returned from the AvaTax web service.
- When to save the results to the AvaTax database in preparation for tax reporting.
- How to handle returns, delayed shipments, credits, and discounts.

Web Service Operations

The main display of the System Control File shows the list of services available to you. This section gives an overview of what each of these services do. It will be up to you to select which ones to use and when to use them in order to develop a system that meets your business requirements.

```

AC001R1                Maintain System Control File                4/16/13
                                                                12:42:47

Type option, press Enter.
 2=Edit 3=Copy 4=Delete 5=Display T=Test

Opt Primary Service      Service Description      Company Code
- Tax Services           GetTax                   APITrialCompany
- Tax Services           PostTax                  APITrialCompany
- Tax Services           CommitTax                APITrialCompany
- Tax Services           CancelTax                APITrialCompany
- Tax Services           AdjustTax                APITrialCompany
- Tax Services           Ping                     APITrialCompany
- Address Services       Validate                 APITrialCompany
- Address Services       Ping                     APITrialCompany

F1=Info  F3=Exit  F6=Add
Copyright 2004-2013 Avalara, Inc. All Rights Reserved

MA  a                MW                07/003

```

Ping

The Ping service for both Tax Services and Address Services provides a means of testing the Internet connection and validating Avalara Credentials (account, license, login, password, etc.). Typical usage is for the connector interface to test connections that appear to be down or unavailable in a new screen or attached to a menu object for the user to call on demand. Typical results would appear as follows:

Ping ResultCode is: Success Ping Version is: 11.11.1.1

The ability to call this service is already provided for you within the System Control File function (see Figure 3.3).

GetTax

Service Code 001

The **GetTax** API is the core of the Avalara web service. It is capable of collecting a string of elements contained within a single object to include origin (ship from address), destination (ship to address), and line items (products, freight, service, etc.) information. Individual line item origin / destination addressing is also available as well, if your order has multiple ship to points.

GetTax can be used in many ways. One of the most important fields in the GetTax call is the document type. This specifies the category of the document and affects how the document is treated after a tax calculation.

GetTax is typically called during a [SalesOrder](#) or [SalesInvoice](#) DocType transaction but here is a full list of document types that are available and a description of what they do.

Document Type	Description
Any	This will return all types of documents.
SalesOrder	Sales Order, estimate, or quote. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of Temporary .
SalesInvoice	The document is a permanent invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of Saved .
PurchaseOrder	Purchase order, estimate, or quote. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of Temporary .
PurchaseInvoice	The document is a permanent invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of Saved .
ReturnOrder	Sales Return Order. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of Temporary .
ReturnInvoice	Sales Return Invoice. The document is a permanent sales return invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of Saved .
InventoryTransferOrder	Inventory Transfer order, estimate, or quote. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of Temporary .
InventoryTransferInvoice	Inventory Transfer Invoice. Document is a permanent invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of Saved .

The return of a successful GetTax call is at a minimum Total Amount and Total Tax. However, by setting the DetailLevel property (for example from “Tax” to “Line”) a much greater set of results is returned.

Here is a full list of DetailLevel types that are available and a description of what they do.

DetailLevel	Description
Summary	Reserved for future use.
Document	Document (GetTaxResult) level details; TaxLines will not be returned.
Line	Line level details (includes Document details). TaxLines will be returned but TaxDetails will not be returned.
Tax	Tax jurisdiction level details (includes Document, TaxLines, and TaxDetails).
Diagnostic	In addition to Tax level details, indicates that the server should return information about how the tax was calculated. Intended for use only while the SDK is in a development environment.

A **GetTaxResult**, when properly coded, would typically appear similar to this:

GetTax is: Success

DocCode: 100811153742

TotalAmount: 1000

TotalTax: 86

Line: 1 Tax: 86 TaxCode: P0000000

Juris Type: State; Juris Name: WASHINGTON; Rate: 0.065000; Amt: 65

Juris Type: City; Juris Name: BAINBRIDGE ISLAND; Rate: 0.021000; Amt: 21

NOTE

- It is not a requirement to call a Validate API prior to a GetTax as a base level address validation as “normalization” occurs as part of the GetTax API function. However, it is a best practices recommendation to validate all addresses first to ensure that a quality calculation will result from passing accurate street, city, state, and zip properties through the GetTax method.*

If you do wish to use SDK adapter to perform address validation before the GetTax call simply switch on the Address Validation flag in the System Control File. See [Configure the System Control File](#).

PostTax / CommitTax

Service Code 003 (Post) and 004 (Commit)

The **PostTax** and **CommitTax** services can be used to set the AvaTax **Document Status** to several states during a **SalesInvoice** process. Remember, **SalesOrder** DocTypes are not saved. AvaTax documents appear initially as “**uncommitted**” on the AvaTax Admin Console and do not appear at all on Avalara reporting until they are “**committed**”.

CancelTax

Service Code 005

This cancels a previously calculated tax. This is for use as a compensating action when posting on the client fails to complete.

This is only available for saved tax document types (Sales Invoices, Purchase Invoices). A document that is saved but not posted is deleted if canceled. A document that has been posted reverts to a saved state if canceled (in this case CancelTax should be called with a CancelCode of PostFailed). A document that has been committed cannot be reverted to a posted state or deleted. In the case that a document on the client side no longer exists, a committed document can be virtually removed by calling CancelTax with a CancelCode of DocDeleted. The record will be retained in history but removed from all reports.

The request must specify all of CompanyCode, DocCode, and DocType in order to uniquely identify the document.

AdjustTax

Service Code 007

Adjust tax allows a user to adjust the invoice on AvaTax. Reconciles tax history to ensure the client data matches the AvaTax history.

Adjustment is allowed for Committed documents. If Document status is **Locked** then system will not process any AdjustTax call. A valid AdjustmentReason is required for adjusting a document.

Interfacing with AvaTax SDK for IBM i

If you are one of our customers who has used or is converting from our SalesTax II product, you are probably aware that the interface between the calculation program TCALC and the connector program is handled by two data structures: PARMSDS and PARMSOP. The SDK adapter uses a very similar approach but uses four data structures: two for input and two for output. The input data structures have one for passing the HTTP and system control parameters and one for passing the order or invoice details. The output structures have one that returns the result and details of the tax calculation and separate structure containing the details of the jurisdictional breakdown, should you require it.

The following is an explanation of the four data structures.

HttpI - HTTP Parameters Used by the Web Service Call

These parameters are easy to populate as the example code shows. Simply chain to the System Control file using a key of Company Code/Service Id and then populate the HTTPPI data structure with the returned results.

Refer to [Appendix A](#) for a table of HTTP parameters, lengths, and types.

Asial – Order/Invoice Parameters

These parameters are used to pass the order or invoice details for which you want a tax calculation or adjustment. Included with your documentation you will find the Microsoft Excel file named **Asial_Mapping_Template**. You can use this tool to map the data fields from your system against those required by the SDK adapter.

Refer to [Appendix B](#) for a table of Order/Invoice parameters, lengths, and types.

NOTE

- *If you are using the V6R1 version of AvaTax SDK for IBM i, then the Address array is increased to 99 elements and the line number array to 999 elements.*

AsiaO – Tax Data Returned by AvaTax

These parameters are returned by the SDK adapter from the AvaTax web service.

Refer to [Appendix C](#) for a table of General Data parameters, lengths, and types.

TxJuO – Tax Jurisdiction Data Returned by AvaTax

These parameters are descriptions of the data structure which returns tax jurisdiction details per order or invoice line. There can be up to 10 separate tax jurisdictions per line (for example, State, County, City, Special, etc.).

Refer to [Appendix D](#) for a table of Tax Jurisdiction Data parameters, lengths, and types.

Required Fields

The following table represents a list of required fields that you need to provide at a minimum to use Avalara web services.

GetTax Members	Type/Size	Required	Description
DocCode	String (50)	Required	Invoice, return, or credit memo number, must be unique at the company level: <i>*Blank field will result in a DocStatusError Message (see appendix B)</i>
DocDate	Date (10)	Required	Invoice, return, credit memo date, Format: <i>*Blank field will result in a InternalServiceFault (see appendix B under "Date")</i>
CustomerCode	Text (50)	Required	Code utilized within your organization's ERP/ eCommerce application to identify a customer. Also used in the Certification process. <i>*Blank field will result in a RequiredError (see appendix B)</i>
LineNo	Text (10)	Required	Transaction line number, must be unique sequence for each document, suggested – sequential number: 1, 2, 3, 4, etc. <i>*Blank field will result in a RequiredError (see appendix B)</i>
ItemCode	Text (50)	Required	Item Code/SKU identifying the product or service being sold to your organization's end-customer. <i>*Blank field will result in a Warning: Missing argument (see appendix B)</i>
Amount	Number	Required	Total sale amount for the line item (Quantity * Unit Price) <i>*Blank field will result in a Warning: Missing argument (see appendix B)</i>
DestAddress	Text (50)	Optional With Caveats	Destination/Ship-to street address. To receive a top quality tax result, the street address must be validate-able. <i>*Blank field will result in a RequiredError (see appendix B)</i>
DestCity	Text (50)	Optional With Caveats	Destination/Ship-to city: To receive a top quality tax result, the City must be validate-able with the Zip Code
DestRegion	Text (2)	Required	Destination/Ship-to state/province 2 character abbreviation <i>*Blank field will result in a TaxAddressError (see appendix B)</i>
DestPostalCode	Text (10)	Required	Destination/Ship-to postal code, five character (12345) or nine character (12345-6789) <i>*Blank field will result in a TaxAddressError (see appendix B)</i>
OrigAddress	Text (50)	Optional With Caveats	Origin/Ship-from street address. To receive a top quality tax result, the street address must be validate-able.
OrigCity	Text (50)	Optional With Caveats	Origin/Ship-from city. To receive a top quality tax result, the City must be validate-able with the Zip Code
OrigRegion	Text (2)	Required	Origin/Ship-from state/province 2 character abbreviation <i>*Blank field will result in a TaxAddressError (see appendix B)</i>
OrigPostalCode	Text (10)	Required	Origin/Ship-from postal code, five character (12345) or nine character (12345-6789) <i>*Blank field will result in a TaxAddressError (see appendix B)</i>

Common System Messages

Here is a list of common system messages you might encounter.

Message	Summary	Documentation
CompanyNotFoundError	Company not found. Verify the CompanyCode.	An unknown CompanyCode was specified. All companies must be configured using the Admin console.
DocStatusError	DocStatus is invalid for this operation.	GetTax expects the document to not exist or the DocStatus to be Saved. PostTax expects the DocStatus to be Saved. CommitTax expects the DocStatus to be Posted.
DocTypeError	DocType is invalid	An unsupported document type was specified. It must be SalesOrder, SalesInvoice, PurchaseOrder, PurchaseInvoice, ReturnOrder, ReturnInvoice
DocumentNotFoundError	The tax document could not be found.	The tax document specified could not be found.
RegionCodeError	Invalid or missing state/province code.	The two-digit state code was not specified or was invalid.
CountryError	Unknown country name or code	The country name or code was not recognized.
TaxAddressError	Address is incomplete or invalid.	An incomplete or invalid origin or destination address was given. RefersTo indicates which. A taxable address must include a (line, city, and region (state)) or a (line and zip) or a (region (state) and postal code).
OutOfBalanceWarning	Document was posted, but is out of balance.	A document was posted in which the specified TotalAmount or TaxAmount did not match the saved document. This is just a warning. The document was posted.
JurisdictionNotFoundError	Unable to determine the taxing jurisdictions.	Unable to determine the taxing jurisdictions.
AddressRangeError	The address number is out of range.	The address was found but the street number in the input address was not between the low and high range of the post office database.
InactiveCompanyError	Tax operations not allowed for an inactive company	Company Code is set to inactive.

DuplicateLineNoError	Duplicate line number	One or more line numbers are duplicates. All line numbers must be unique within the document.
TaxRegionError	The TaxRegionId was not found	
TaxOverrideError	Tax override cannot be applied.	
AddressError	Unable to validate the address.	
InsufficientAddressError	Insufficient address information	You must specify at least Line/ZIP, or Line/City/State
PostalCodeError	Invalid ZIP/Postal Code.	The ZIP/Postal Code does not exist and could not be determined by the city/municipality and state/province.
UnsupportedCountryError	Country not supported.	Address Validation for this country not supported.
Messages.DateRangeError	Start Date cannot be later than the End Date	An invalid date range was specified. RefersTo indicates the property.
Messages.RequiredError	Required element is missing	A required property was not provided. RefersTo indicates which property. This value must be specified.
Messages.UniqueConstraintError	This is a duplicate of an existing unique element and cannot be saved.	Please verify the element is unique.
Messages.MaxCountExceededError	MaxCount limit exceeded. Please use filters to limit the result set.	1000 invoice lines is the current limit.

Coding Considerations

Let's start off with a pseudo-code example of a typical GetTax scenario.

In this particular system, the Invoice Header has a status of 'Shipped' when it is ready for invoicing and a Ship To and Ship From addresses are specified at the Invoice Header level.

Retrieve the SDK adapter System Control details by Company/Service Key (A);

If found then;

Clear Httpi buffer;

Populate the Httpi data structure;

Retrieve the first Invoice Header by status '*Shipped*' (B);

Repeat until no more '*Shipped*' headers;

Clear the SDK adapter buffers;

Address No = 0;

Line No =0;

Populate the document level details in the Asial data structure (C);

Address No = Address No +1;

Populate the Asial Address details for Ship To;

Address No = Address No +1;

Populate the Asial Address details for Ship From;

Read the first Invoice Line (D);

Repeat until no more lines for this Invoice;

Line No = Line No +1;

Populate Asial line level details;

Read next Invoice Line;

Call the Avalara GetTax Service to get the tax details (D);

If Return Code = Error;

Take some error action;

else;

Update the Invoice Header/Details with details returned in AsiaO (E);

end;

Read next Invoice Header by status '*Shipped*';

else;

Take some error action;

end;

There is a sample program (not designed to compile) which creates the above example in free format RPG. You can find this in AVATAX/QRPGLESRC GETTAX_EXP. The lines above that have letters at the end (for example (A)) you can search for and find in the RPG example.

Also, included in the QRPGLESRC file you will find (working) source for the tax rate lookup, tax rate test, and address validation programs that came with your installation (try them out on the Inquiry menu). These may be helpful to you and are named as follows:

Member	Type	Description
AX002R1	RPGLE	SDK adapter - Tax Rate Look Up
AX003R1	RPGLE	SDK adapter - Tax Rate Look Up - Ship From/To
AX005R1	RPGLE	SDK adapter - Address Validation Tester

Other Considerations

GetTax

Depending on your business model you may find that the **GetTax** call is all you need. If you are or have ever been a SalesTax II customer, you can use this in the same way as you would that program (that is you can commit the document to the AvaTax Admin Console at the same time as you retrieve the tax calculation). This is just like how SalesTax II calculates the tax and writes the details to the TAXACCUM file at the same time.

To do this you will need to use a document type of 'SalesInvoice' or 'PurchaseInvoice' and set the commit flag to 'true'.

NOTE

- *Use with care! Once a document is committed it cannot be modified other than with **CancelTax**. (see [AvaTax Document States](#)).*

PostTax / CommitTax, CancelTax, AdjustTax

If you use these functions you can either call them using the DocId originally passed back from the GetTax call (if you saved it) or you can call them or using your own document code and company code, just as you did when you called GetTax.

Detail Level Enumeration

In a `GetTaxRequest` (or `GetTaxHistoryRequest`) the `DetailLevel` specifies the level of tax detail to return to the client application following a tax calculation.

DetailLevel	Description
Summary	Reserved for future use.
Document	Document (<code>GetTaxResult</code>) level details; <code>TaxLines</code> will not be returned.
Line	Line level details (includes Document details). <code>TaxLines</code> will be returned but <code>TaxDetails</code> will not be returned.
Tax	Tax jurisdiction level details (includes Document, <code>TaxLines</code> , and <code>TaxDetails</code>).

How results would appear depends on the `DetailLevel` set during the `GetTax` Call.

Document

GetTax Result: Success

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

Line

GetTax Result: Success

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

Line: 1 Amount: 1000 Tax: 83.5

ItemCode: AvalItem-001 TaxCode: P0000000

Tax

GetTax Result: Success

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

Line: 1 Amount: 1000 Tax: 83.5

ItemCode: AvalItem-001 TaxCode: P0000000

Juris Type: State; Juris Name: COLORADO; Rate: 0.029000; Amt: 29

Juris Type: County; Juris Name: ADAMS; Rate: 0.007500; Amt: 7.5

Juris Type: City; Juris Name: COMMERCE CITY; Rate: 0.035000; Amt: 35

Juris Type: Special; Juris Name: SCIENTIFIC & CULTURAL FAC.(CD); Rate: 0.001000; Amt: 1

Juris Type: Special; Juris Name: METRO FOOTBALL STADIUM DISTRICT (FD); Rate: 0.001000; Amt: 1

Juris Type: Special; Juris Name: RTD GREATER DENVER; Rate: 0.010000; Amt: 10

Where the DetailLevel setting becomes very important is in the tax details that are returned from countries that make use of GST, PST, and HST – for example Canada.

GST = General Sales Tax (appears in AvaTax CNT or Country)

PST = Provincial Sales Tax (appears in AvaTax State)

HST = Harmonized Sales Tax (State - combined similar to SST)

VAT = Value Added Tax (appears in AvaTax State)

These values would be returned in the "Tax" DetailLevel as follows:

Tax

GetTax Result: Success

Address: 1919 Upper Water St, Halifax, NS B3J 3J5

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

Line: 1 Amount: 1000 Tax: 151.00

ItemCode: Avaltem-001 TaxCode: P0000000

Juris Type: State; Juris Name: NOVA SCOTIA; Rate: 0.10; Amt: 100 (or PST / HST / VAT)

Juris Type: County; Juris Name: CANADA; Rate: 0.050; Amt: 51 (or GST)

AvaTax Document States

Saved - The last call was to GetTax, or the last call was to CancelTax with the PostFailed flag set, or the document was in a state of Posted when a CancelTax was called.

Posted - The last call was to PostTax.

Committed - The last call was to CommitTax.

NOTE

- *Only invoices in a Committed state appear on reports.*

Voided - The last call was to CancelTax with DocDeleted flag set.

Therefore:

- For a **GetTax** to succeed, the document must either not exist (a new invoice) or be in a state of saved. If the document is in any other state, the ResultCode will be Error, and the Message will be "DocStatus is invalid for this operation". Successful calls leave the document in a saved state.
- For **PostTax** to succeed, the document must be in a state of saved. A successful call leaves the document in a posted state.
- For **CommitTax** to succeed, the document must be in a state of posted. A successful call leaves the document in a committed state.

- For **CancelTax** with **PostFailed** flag to succeed, the document must be in a state of posted. A successful call leaves the document in a saved state.
- For **CancelTax** with **DocDeleted** flag to succeed, the document can be in any state except voided (saved, posted, committed). A successful call leaves the invoice in a voided state.

NOTE

- *There is no way to move an invoice from a state of voided to any other state.*

Handling Return Invoices

Return Invoice processing is largely a business practice that the developer and the business manager need to map out prior to coding or moving forward to a production environment. As the business needs of every company is as unique as their products, it should not be expected that the scenarios described below will work for your specific situation, or that we have any “one size fits all” solution that you can “snap in” to your development; rather it is up to the developer to fashion a solution that fits the company’s business model.

Committed Document – Refund (Return Invoice)

For the example provided below, we will process a return invoice on the following invoice.

Document No	Customer / Vendor Code	Batch Code	Document Type	Document Date	Cur	Total	Non-Taxable	Taxable	Tax Location Code	Document Status	Status Date
70000	AvaCustomer		Sales Invoice	7/19/2010	USD	3375.00	0.00	3375.00	163.71	Committed	8/19/2010

The invoice has the following line items:

No	Item	Tax Code	Use Code	Sourcing	Line Amount	Discount	Non-Taxable	Taxable	Tax
1	NBCP-000-S	PC030100		D	100.00	0.00	0.00	100.00	4.85
2	NBCP-000-L	PC030100		D	125.00	0.00	0.00	125.00	6.08
3	NBCP-000-2XL	PC030100		D	500.00	0.00	0.00	500.00	24.25
4	NBCP-000-3XL	PC030100		D	800.00	0.00	0.00	800.00	38.80
5	NBBP-000-L	PC030100		D	1800.00	0.00	0.00	1800.00	87.30
6	Shipping	FR020200		D	50.00	0.00	0.00	50.00	2.43

Assumptions:

- The document in question has already been **committed** and tax remitted to the tax jurisdictions.
- There may be multiple lines in the document.
- A **complete** or **partial** refund is the expected outcome

Steps:

1. Call a GetTax with a duplicate of the document (invoice) you want to process returns on:
 - a. using the same invoice number, or
 - b. with a new invoice number (DocCode) with the original invoice number passed in the reference number field, or
 - c. re-use the original document’s invoice number with a “.1” added to it

NOTE

- *You can only use ReturnInvoice method once using the original DocCode. A DocStatusError results if you attempt to commit a second document using ReturnInvoice with a duplicate DocCode*

Document No	Customer / Vendor Code	Batch Code	Document Type	Document Date	Cur	Total	Non-Taxable	Taxable	Tax	Location Code	Document Status	Status Date
70000	AvaCustomer		Sales Invoice	7/19/2010	USD	3375.00	0.00	3375.00	163.71		Committed	8/19/2010
70000.1	AvaCustomer		Return Invoice	8/19/2010	USD	-600.00	0.00	-600.00	-29.10		Committed	8/19/2010

- Set the DocType to ReturnInvoice (very important if you are re-using the original DocCode).
- Set the DocDate to the tax reporting month that you want the return to appear in (typically the current month).
- Set the TaxOverride property to TaxDate.
- Set the TaxOverrideDate to the date the original invoice computed tax. AvaTax calculates the tax based on this date.
- Set the AdjustmentReason to Return Items (or whatever makes sense for your business needs).
- Optional: Set the GetTaxRequest **ReferenceCode** = to the original Invoice for tracking purposes.

NOTE

- Pass only line items being returned. Do not include the line items that will not be returned unless all items are returned. In the example below we are returning line items 1 and 3.

- Set the Amt field to a negative dollar amount.

NOTE

- Always leave Qty as a positive.

No	Item	Tax Code	Use Code	Sourcing	Line Amount	Discount	Non-Taxable	Taxable	Tax
1	NBCP-000-S	PC030100		D	-100.00	0.00	0.00	-100.00	-4.85
3	NBCP-000-2XL	PC030100		D	-500.00	0.00	0.00	-500.00	-24.25

Once sent to the AvaTax web service, the tax engine will return negative tax amounts on the line items based on the TaxDate specified. If no TaxDate is set, the document date will be used to calculate tax.

NOTE

The Taxable Amounts on a Return Invoice show negative amounts equal to the items returned.

Summary:

Out of the six items originally processed, two items have been reversed on your tax reporting – in this case \$29.10 appears essentially as a “credit” on the current month’s tax liability.

Handling Shipping and Freight

Shipping (Freight) should be sent as a separate line item on Documents (invoice). Use the pre-defined tax code of FR020100 (GetTaxRequest TaxLines(x) TaxCode). This tax code will automatically charge or not charge tax on Freight, according to the regulations of the state referenced in the Ship To address.

Handling Discounts

The recommended best practice for handling discounts is in the following manner:

- It is expected that the GetTax call passes a dollar amount of the discount at the Document level.
- It is also expected that anywhere from one to many line items have the Discounted property set to 'true'.
- For each line item to which the discount applies AvaTax allocates the discount across those line items and reduces the tax base by the amount of the allocation.
- On the AvaTax Admin Console, the line items show the discounted amounts that applied (see Figure 4.1).
- For line items that you want discounted, ensure you pass the net amount (versus the discounted amount) to the Adapter.

NOTE

- *Manufacturing Discounts should not be passed to AvaTax.*

Some Examples:

1. **Basic:** A \$25.00 line item is sold with a \$10.00 discount. The resulting tax base is \$15.00.
Simple: Two \$25.00 line items are sold with a \$10.00 discount, with only one item marked for the discount. The resulting tax base is \$15.00 for the item marked for Discount and \$25.00 for the item not marked for discount.
2. **Typical:** Two \$25.00 line items are sold with a \$10.00 discount, with both line items marked for the discount. The resulting tax base is \$20.00 for each item marked for discount.
3. **Complex:** Three line items and a Freight line item are sold (one for \$25, \$35, and \$45) with a **\$20.00** discount, with all items (except Freight) marked for the discount. The resulting tax calculation would appear this way on the Admin Console.

Origin: 100 RAVINE LN NE, BAINBRIDGE ISLAND, WA, 98110-2687, US					Destination: 100 RAVINE LN NE, BAINBRIDGE ISLAND, WA, 98110-2687, US				
Document ID:	17530654	Salesperson Code:							
Document Code:	102111-1	Entity Use Code:							
Customer/Vendor Code:	AvaTim	Total Amount:						120.00	
Document Status:	Uncommitted on 10/21/2011	Total Discount:						20.00	
Document Date:	10/21/2011	Total Non-Taxable:						0.00	
Payment Date:	1/1/1900	Total Taxable:						100.00	
Document Type:	Sales Invoice	Tax Override Type:						None	
Location Code:		Tax Override Reason:							
Tax Date:	10/21/2011	Total Tax:						8.60	
Purchase Order No:		Adjustment Reason:						Not Adjusted	
Reference Code:		Adjustment Description:							

☐ Expand All
 Currency in USD

No	Item	Tax Code	Use Code	Sourcing	Line Amount	Discount	Non-Taxable	Taxable	Tax	TO	JO	Tax Included
1	AvaProduct 1	P0000000		D	25.00	4.76	0.00	20.24	1.75			
2	AvaProduct 2	P0000000		D	35.00	6.67	0.00	28.33	2.43			
3	AvaProduct 3	P0000000		D	45.00	8.57	0.00	36.43	3.12			
4	FREIGHT	FR020200		D	15.00	0.00	0.00	15.00	1.30			

Ready

Figure 4.1

NOTE

- *The discount was apportioned across the three items but not the Freight.*
- *The taxable amount was reduced by a total of the discount applied (\$20).*

Messages – Results / Error

Understanding and taking appropriate action on error messages is **critical** to the successful implementation of a custom SDK connector. The errors and warnings that result from web service calls guide the receiver to the most logical problem resolution path.

For example; if an API, such as GetTax fails, the receiver should always iterate through the messages collection that is part of the result set and check the contents of the Name, Summary, Details, and Refers To fields of each message. Often the contents of these messages will indicate what the problem is.

For example, below is a SOAP trace from a DocumentNotFoundError:

```
<Messages>
  <Message Name="DocumentNotFoundError">
    <Summary>The tax document could not be found.</Summary>
    <Details>SO123456789</Details>
    <RefersTo />
    <Severity>Error</Severity>
    <Source>Avalara.AvaTax.Services.Tax</Source>
  </Message>
</Messages>
```

All connectors should check the ResultCode on the return from the web service call and contain code that will program for:

- ResultCode = SeverityLevel.**Success**
- ResultCode = SeverityLevel.**Warning**
- ResultCode = SeverityLevel.**Error**
- ResultCode = SeverityLevel.**Exception**

NOTES

- Typically, **Exception** is only returned by a system error from the Avalara web service and will be caused by circumstances outside of the client application's control (that is not having an Internet connection will return the error).
- In general, the path taken for ResultCode= **Exception** should be the same as for ResultCode=**Error**.
- The only known circumstance under which ResultCode=**Warning** would be returned is for the API PostTax when the TotalAmt or TotalTax fields do not agree with the amounts in the Avalara database.
- At this time, the only API that may return SeverityLevel.Warning is PostTax.

Address Validation - GetTax Method

While a fully validated street address is always better, taxing jurisdictions can usually be unambiguously determined if at a minimum city, state, and ZIP Code are provided (and at least two-thirds are valid).

However, even though it is possible for the **GetTax** request to be called devoid of a valid street address or even ZIP Code, the **GetTaxResults** may often omit Special taxes (that is transit, stadium, levy, etc.) that are specific to the jurisdictional boundaries the shipped to address may be liable for. Without a valid street address it may be next to impossible to return tax results that should be included if the address was less ambiguous.

A **GetTax** call attempts to normalize an address, but does not return the normalized address as part of **GetTaxResults**. If the address cannot be normalized, it continues with processing a Sales Order/Invoice with what it has. Albeit rare, incomplete or erroneous addresses submitted via **GetTax** operations will return incorrect results.

Setting the Address Validation flag to 'Y' in the system control calls the address validation service to 'cleanse' any addresses before calling **GetTax**.

Understanding Boundary levels

Boundary levels, returned in **GetTaxResults**, indicate the jurisdiction boundary precision level enumerations (below) found for the address submitted. From highest (1) to lowest (3):

1. Address: Street address precision
2. Zip9: 9-digit zip precision
3. Zip5: 5-digit zip precision

The result depends on the accuracy of the address as well as the precision level of the state provided jurisdiction boundaries.

NOTE

- *In some jurisdictions, the highest level of precision to return an accurate tax result may well be level two or three. However, this is the exception versus the rule.*

Handling Tax Jurisdiction Breakdowns

You will remember that the tax breakdown by jurisdiction is returned in the TxJuO data structure. This data structure is multi-dimensional. There is an element for each line item on the document and up ten elements within that for each taxing jurisdiction. In your RPG code you would retrieve that information similar to the example shown in Figure 4.2 below.

```
//  
// Read all the Tax Jurisdictions for an Invoice Line  
//  
line = %int(O_No);  
jur = 1;  
do (TxJuO(line).O_TaxName(jur) = *Blanks) or (jur > 10);  
  TaxName = TxJuO(line).O_TaxName(jur);  
  JurName = TxJuO(line).O_JurisName(jur);  
  TaxRate = TxJuO(line).O_Rate(jur)*100;  
  eval(H) TaxAmt = TxJuO(line).O_Tax(jur);  
  jur = jur + 1;  
enddo;
```

Figure 4.2

Useful Links

WSDLs

- You won't actually need to use these directly, the SDK adapter does it for you but they may be useful to get a better understanding of the service.

Development

<https://development.avalara.net/Tax/TaxSvc.wsdl>

<https://development.avalara.net/address/addresssvc.wsdl>

Production

<https://avatax.avalara.net/address/addresssvc.wsdl>

<https://avatax.avalara.net/Tax/TaxSvc.wsdl>

Avalara Admin Console

These are links to the development and production AvaTax Admin Consoles.

Development

<https://admin-development.avalara.net/>

Production

<https://admin-avatax.avalara.net/>

Web Service URLs

These are the development and production web service URLs.

Development

<https://development.avalara.net>

Production

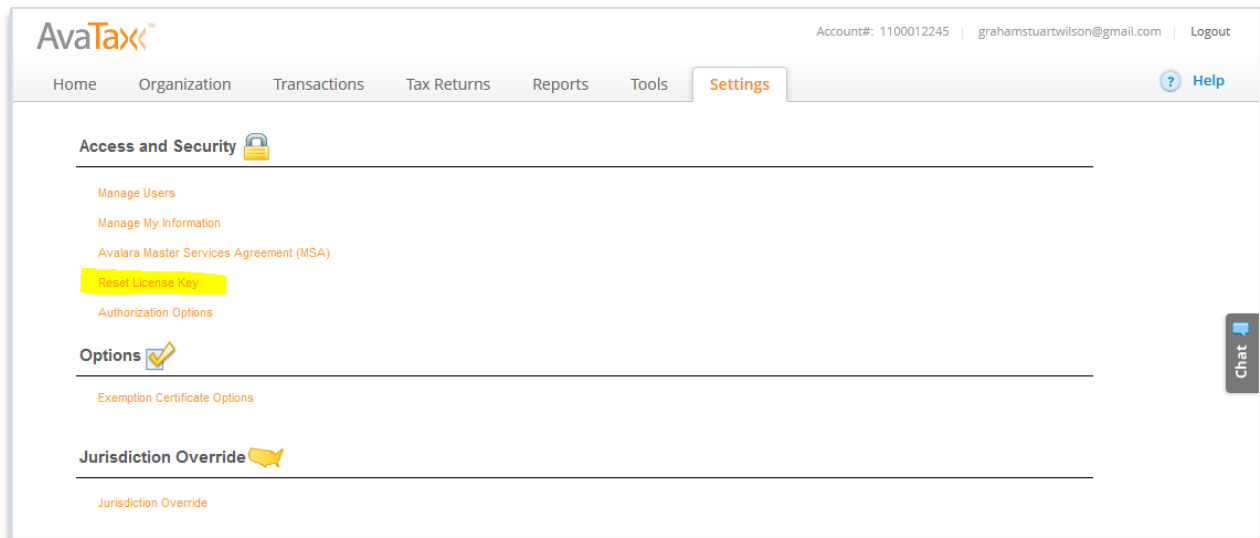
<https://avatax.avalara.net>

developer.avalara.com

You will find useful links and information here to get you building your SDK adapter connector. Use the free trial to get an Admin Account (pre-set up) that you can use while developing your SDK adapter connector. Trial accounts are valid for one month.

Resetting Your License Key

If you've lost or forgotten your license key you can reset it. Sign in to the AvaTax Admin Console. Select the Settings tab and then click Reset License Key.



NOTE

- *Resetting the license key has the effect of invalidating the old key, and breaks all connector connection to the AvaTax web service still using that old license key (including AvaTax ERP Connectors using the same account number). In other words, when you reset your license key, you will need to change the credential properties in the AvaTax SDK for IBM i System Control file for all services using the same account number.*

Tax Compliance

AvaTax tax calculations are “data-driven”, meaning data that affects a particular tax calculation may affect:

- State and Local Nexus Admin Console settings
- Taxability rules based upon Customer type codes and Item tax codes
- Exemption Certificates
- Other tax rules defined

NOTE

- *SST and AvaTax Returns customers are required to send a value in the; `GetTax Lines[n].ItemCode` and `Description` fields.*

Nexus Jurisdictions

Nexus Jurisdictions (formerly Nexus) is a term meaning the sufficient connection a business has with a taxing jurisdiction. This connection obligates the business to calculate, collect, report, and remit tax. The connection is established by virtue of the business activity conducted in the taxing jurisdiction.

Typically, a business in the United States must have a “substantial physical presence” in a taxing jurisdiction to be required to register there. The following may constitute a “substantial physical presence” in a taxing jurisdiction, and therefore an obligation to calculate, collect, report, and remit tax:

- A corporate office, storefront, or remote sales office
- Remote employees working from home on business payroll
- Owned or rented warehouse space containing owned inventory
- Owned inventory leased to a customer
- Sales or marketing representatives making regularly scheduled visits
- Delivery of product by a business owned vehicle

Rules governing what constitutes “substantial physical presence” vary from jurisdiction to jurisdiction. Determining exactly how a rule applies to your business is critical.

Businesses in the European Union may not have a “substantial physical presence” in an EU country. Instead, businesses can select other EU countries where they:

- Have reached or exceeded the buying or selling threshold of that EU country, or
- Have voluntarily decided to register in that EU country to reclaim VAT charges

NOTE

- *Contact Avalara Professional Services, an accountant, a tax attorney, or other qualified tax professional to conduct a Nexus Jurisdictions analysis for your US-based business. Avalara Professional Services offers Nexus Jurisdictions analysis only for US-based businesses at this time.*

For US-based businesses, AvaTax Calc assumes:

- Transactions in Nexus Jurisdictions are fully taxable
- Transactions in non-Nexus Jurisdictions are non-taxable and receive a 0.00% tax calculation

For EU-based businesses, the EU country in which the transaction takes place is only one consideration that AvaTax Calc uses. Other factors affecting AvaTax Calc are the type of good sold, whether that good is taxable or exempt, and whether it is sold to a business or a consumer.

More granular control over taxing Transactions is provided by

- Tax Codes
- Tax Rules
- Items

If you have more questions, please refer to your Admin Console setup or check out the AvaTax Admin Console Training.

ItemCode versus TaxCode

ItemCode versus TaxCode in line Items:

- Special product taxability rules can be defined using Item and/or TaxCodes:
 - Line.ItemCode or Line.TaxCode
- ItemCodes represent individual products, whereas TaxCodes represent categories of products. For example:
 - ItemCodes of M for Milk, E for Eggs, and C for Cheese.
 - Then a TaxCode of GR might be established for the broader category of Groceries.

Product Taxability Rules are defined in the AvaTax Admin Console in terms of TaxCodes.

For example: Groceries are not taxable in the state of Washington so a taxability rule might be established that states 'when a line item is passed with the Tax Code GR, and the destination address is Washington, the line item is charged \$0 tax'.

The mechanism to get the tax rule defined above by one of two methods:

1. In the AvaTax Admin Console, map ItemCodes to a TaxCode(s) then send the ItemCode for your individual product SKU.
 - a. Using the example above, ItemCodes M, E, and C would be mapped to each via TaxCode GR. Then if Line.ItemCode = M, E or C they would automatically map to TaxCode GR, and the rule for TaxCode GR will apply.
2. Send the TaxCode GR in Line.TaxCode field.
 - a. Method #1 is popular, because it allows you to send your own internal SKUs without setting up any additional tables. This is especially true for PRO or SST customers because Avalara is typically supplying the TaxCode(s) for them.

If you send a value in **both** the ItemCode and the TaxCode fields, the behavior is as follows:

1. If the **ItemCode** sent maps to a **TaxCode**, the mapped TaxCode will be used for product taxability rules, regardless of what is sent in the TaxCode field.
2. If the **ItemCode** sent does not map to a **TaxCode**, the TaxCode sent will be used for product taxability rules.
3. Lines that have, or are mapped to **TaxCodes** that have no rule associated with them are simply taxed at the applicable rates for Origin, Destination and Nexus Jurisdiction settings.

Pre - GoLive Check List

Once you have things up and running, here is a recommended checklist you should go through in order to validate that your connector is working correctly.

Documents

- Evidence of committed documents in the AvaTax Admin Console.
- Evidence that a Freight/Shipping line was tested within a saved document.
- Evidence of a voided document (use of **CancelTax**).
- Evidence that returns, partial returns, delayed shipping, and if appropriate drop shipping methods were tested successfully.
- Evidence that Return Invoices are passing negative amounts versus negative quantities.

Item Codes and Description

- Verification that there are Item Codes with corresponding Item Descriptions (a requirement for SST and AvaTax Returns customers).
- Verification that tax rules, tax codes, and / or Item Codes mapped to tax codes are tested in transactions and are functioning as expected. Those that appear out of character will be flagged and the developer notified via the analysis process.
- Verify that all Items Codes are not set to the default tax codes. For example; item codes mapped to Tangible Personal Property or P0000000.
- Verify that the use of the FR tax code (Freight) is not defaulting to the Tangible Personal Property (P0000000) tax code which may have the effect of incorrect tax results.

Nexus Jurisdictions

- Verify the company's local nexus selected (the tax jurisdiction they are doing business in).
- Verify that the Nexus Jurisdiction settings appear to support the expressed or written business needs of the customer.
- Evidence that you have tested transactions that fall inside and outside of Nexus Jurisdictions.

Server Audit

- Evidence of the use of DocType="SalesOrder" is present and results in a saved document (DocType="SalesInvoice").
- An analysis that results in an "appropriate" ratio of Validate and SalesOrder to SalesInvoice transactions. Best practice recommendation is ratio to be 2-6-1 (that is 2 address validations and 6 SalesOrder transactions to complete 1 committed document [reportable]).
- An analysis that results in an "appropriate" number of errors (that is address validation errors, or tax service errors encountered in a normal business flow).

Verification that future production ready company codes are not flagged as Test companies. Test companies will be reported as part of the analysis results.

Appendix A: HTTP Parameters

Parameter Name	Length	Type	Dec	Array	Description	Field name in AX0001P	I/O Usage	Required or Optional
Service	3	Char			Key to the service as defined on the System Control file Id: field (for example GetTax = '001')	CTL_SRVKEY	Input	Required
Username	25	Char			The Acct: field from the System Control	CTL_USER	Input	Required
Password	20	Char			The Pass: field from the System Control	CTL_PASS	Input	Required
Port	5	Int	0		The Port: field from the System Control	CTL_PORT	Input	Required
EndPoint	75	Char			The End Point: field from the System Control	CTL_ENDPNT	Input	Required
Name	25	Char			The Service Provider: field from the System Control	CTL_SRVNAM	Input	Required
Client	25	Char			The ERP Ver: field from the System Control	CTL_CLIENT	Input	Required
Adapter	25	Char			The Adapter Ver: field from the System Control	CTL_ADAPTR	Input	Required
Machine	25	Char			The Client System: field from the System Control	CTL_MACHNE	Input	Required
NumHdrs	3	Int	0		The No. Hdrs: field from the System Control	CTL_NBRHDR	Input	Required
Header	30	Char		Dim(3)	The HTTP Header: field from the System Control	CTL_HDR1,2,3	Input	Required
HeaderData	75	Char		Dim(3)	The HTTP Data: field from the System Control	CTL_DTA1,2,3	Input	Required
ValidAddr	1	Char			The Address Validation flag: field from the System Control	CTL_VALADR	Input	Required
TextCase	10	Char			The Address Conversion type: field from the System Control	CTL_TXTCSE	Input	Required if CTL_ADDR = 'Y'
RemoveXML	1	Char			The Remove IFS Files: field from the System Control	CTL_RMVXML	Input	Optional
DebugFlag	1	Char			The Debug: field from the System Control	CTL_DEBUG	Input	Optional
Message	50	Char			Generic message field		Output	
ErrCode	10	Char			Error code returned from the service		Output	
ErrSeverity	10	Int	0		Severity of the error		Output	
ErrPgm	30	Char			The program or routine that trapped the error		Output	
ErrText	128	Char			Text about the error		Output	

Appendix B: Order/Invoice Parameters

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
For a Document, use the following parameters:							
I_Company	25	Char			Your company code as specified in the Company field of the AvaTax SDK for IBM i System Control file.	Input	Required
I_DocType	25	Char			The document type specifies the category of the document and affects how the document is treated after a tax calculation. See DocumentType for more information about the specific document types. See GetTax for details of how this can be populated.	Input	Required
I_DocCode	50	Char			The internal reference code used by your ERP to identify the document (for example, Invoice # or Order #).	Input	Optional
I_NewDocCode	50	Char			The DocCode can be changed during commit using NewDocCode by entering the new code here.	Input	Optional
I_DocID	10	Char			Unique ID returned by GetTax. Can be used as an input to PostTax, CommitTax, CancelTax or AdjustTax.	Both	Optional
I_DocDate	10	Char			The date on the invoice, purchase order, etc. in YYYY-MM-DD format.	Input	Required
I_SalesPsn	25	Char			The client application salesperson reference code.	Input	Optional
I_CusCode	50	Char			The customer code associated with a certificate record.	Input	Optional
I_CusUseTy	25	Char			The client application customer or usage type. The standard values for the CustomerUsageType (A-L) are: A - Federal Government B - State/Local Govt. C - Tribal Government D - Foreign Diplomat E - Charitable Organization F - Religious/Education G - Resale H - Agricultural Production I - Industrial Prod/Mfg. J - Direct Pay Permit K - Direct Mail L - Other	Input	Optional
I_Discount	15	Dec	4		The discount amount to apply to the document.	Input	Required
I_PurchOrd	50	Char			Purchase Order Number for this document.	Input	Optional
I_ExemptNo	25	Char			Exemption Number for this document.	Input	Optional
I_OrigCode	10	Char			User assigned code to indicate that an address is to be used as the "Ship From" location for a specific line item. This can be specified at the document or line detail.	Input	Optional
I_DestCode	10	Char			User assigned code to indicate that an address is to be used as the "Ship To" location for a specific line item. This can be specified at the document or line detail.	Input	Optional
I_DetailLvl	10	Char			Specifies the level of detail to return (see GetTax)	Input	Required

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
I_RefCode	10	Char			For returns (see DocumentType), refers to the DocCode of the original invoice.	Input	Optional
I_HashCode	10	Int	0		Hash Code. You can set to 0.	Input	Required
I_LocCode	10	Char			Also referred to as a Store Location, Outlet Id, or Outlet code. Contains a number assigned by the State which identifies a Store location. Some state returns require taxes are broken out separately for Store Locations.	Input	Optional
I_Commit	5	Char			This has been defaulted to false. Invoice will be committed if this flag has been set to true.	Input	Required
I_BatchCode	10	Char			Optional Batch Code	Input	Optional
I_TaxOrideT	25	Char			Type of TaxOverride. None – Default TaxAmount - The TaxAmount overrides the total tax for the document. Exemption - Exemption certificates are overridden making the document taxable. TaxDate - The TaxDate overrides the DocDate as the effective date used for tax calculation. AccruedTaxAmount - The accrued tax amount	Input	Optional
I_TaxAmt	15	Dec	4		The TaxAmount overrides the total tax for the document.	Input	Optional
I_TaxDate	10	Char			Tax Date is the date used to calculate tax on the Document in YYYY-MM-DD format.	Input	Optional
I_Reason	25	Char			Reason for applying TaxOverride.	Input	Optional
I_Currency	3	Char			3 character ISO 4217 currency code (for example, USD)	Input	Optional
I_ServMode	10	Char			It provides the ability to controls whether tax is calculated locally or remotely when using an AvaLocal server. The default is Automatic which calculates locally unless remote is necessary for non-local addresses. Automatic - Automated handling by local and/or remote server. Local - AvaLocal server only. Lines requiring remote will not be calculated. Remote - All lines are calculated by AvaTax remote server.	Input	Required
I_PayDate	10	Char			Indicates the date payment was applied to this invoice.	Input	Required
I_ExchgRate	15	Dec	4		Indicates the currency exchange rate from the transaction currency (indicated by CurrencyCode) to the company base currency. This only needs to be set if the transaction currency is different than the company base currency. It defaults to 1.0.	Input	Required
I_ExRteEff	10	Char			Indicates the effective date of the exchange rate. It should be set in conjunction with I_ExchgRate. It will default to the I_DocDate if not set.	Input	Required
I_PosLaneCde	10	Char			Optional POS Lane Code	Input	Optional

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
I_BINH	9	Char			Optional Business Identification Number	Input	Optional
I_CancelCode	25	Char			Used by CancelTax . A code indicating the reason the document is getting canceled. Valid values are: PostFailed DocDeleted DocVoided AdjustmentCancelled	Input	Required
I_TotalAmt	15	Dec	6		Used by PostTax . The sum of all line Amount values.	Input	Required
I_TotalTax	15	Dec	6		Used by PostTax . The total tax for the document.	Input	Required
I_AdjReason	3	Int	0		Used by AdjustTax . Sets a valid reason for the given AdjustTax call. Adjustment Reason is a high level classification of why an Original Document is being modified. 0 Not Adjusted 1 Sourcing Issue 2 Reconciled with General Ledger 3 Exemption Certificate Applied 4 Price or Quantity Adjusted 5 Item Returned 6 Item Exchanged 7 Bad Debt 8 Other (Explain)	Input	Required
I_AdjDesc	25	Char			Used by AdjustTax . Sets description for the given AdjustTax call. Adjustment Description is required when AdjustmentReason is "Other" for enhanced tractability.	Input	Optional
For Addresses, use the following parameters:							
I_AddrCode	3	Char		Dim(15)	Programmatically determined value used internally by the adapter. Defaults to the hash code of the Address object. This is Origin or Destination code defined at the document header or line detail level.	Input	Optional
I_Line1	50	Char		Dim(15)	Address line 1	Input	Optional
I_Line2	50	Char		Dim(15)	Address line 2	Input	Optional
I_Line3	50	Char		Dim(15)	Address line 3	Input	Optional
I_City	50	Char		Dim(15)	City	Input	Optional
I_Region	2	Char		Dim(15)	State or province name or abbreviation	Input	Optional
I_PostCode	10	Char		Dim(15)	Postal or ZIP Code	Input	Optional
I_Country	2	Char		Dim(15)	Country Code	Input	Optional
I_TaxRegId	10	Int	0	Dim(15)	TaxRegionId provides the ability to override the tax region assignment for an address.	Input	Required
I_Latitude	10	Char		Dim(15)	Latitude	Input	Optional
I_Longitude	10	Char		Dim(15)	Longitude	Input	Optional
For Line Details, use the following parameters:							

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
I_No	10	Char		Dim(100)	Line #	Input	Optional
I_Origin	10	Char		Dim(100)	User assigned code to indicate that an address is to be used as the "Ship From" location for a specific line item. This can be specified at the document or line detail.	Input	Optional
I_Dest	10	Char		Dim(100)	User assigned code to indicate that an address is to be used as the "Ship To" location for a specific line item. This can be specified at the document or line detail.	Input	Optional
I_Item	50	Char		Dim(100)	Item Code (SKU)	Input	Optional
I_TaxCode	25	Char		Dim(100)	System or Custom Tax Code.	Input	Optional
I_Qty	15	Dec	4	Dim(100)	The quantity represented by this line.	Input	Required
I_Amt	15	Dec	4	Dim(100)	The total amount for this line item (Qty * UnitPrice).	Input	Required
I_Disc	5	Char		Dim(100)	Is this line item discounted? 'true' or 'false'	Input	Required
I_RevAcct	50	Char		Dim(100)	Revenue Account	Input	Optional
I_Ref1	50	Char		Dim(100)	Client specific reference field	Input	Optional
I_Ref2	50	Char		Dim(100)	Client specific reference field	Input	Optional
I_Exempt	25	Char		Dim(100)	Exemption Number for this document	Input	Optional
I_CustUse	25	Char		Dim(100)	The client application customer or usage type. The standard values for the CustomerUsageType (A-L) are: A - Federal Government B - State/Local Govt. C - Tribal Government D - Foreign Diplomat E - Charitable Organization F - Religious/Education G - Resale H - Agricultural Production I - Industrial Prod/Mfg. J - Direct Pay Permit K - Direct Mail L - Other	Input	Optional
I_Desc	200	Char		Dim(100)	Defines the description for the product or item	Input	Optional
I_OrideTyp	25	Char		Dim(100)	Type of TaxOverride. None - Default TaxAmount - The TaxAmount overrides the total tax for the document. Exemption - Exemption certificates are overridden making the document taxable. TaxDate - The TaxDate overrides the DocDate as the effective date used for tax calculation. AccruedTaxAmount - The accrued tax amount.	Input	Optional
I_OrideAmt	15	Dec	4	Dim(100)	Tax override amount	Input	Optional
I_OrideDat	10	Char		Dim(100)	The TaxDate overrides the document date (I_DocDate) as the effective date used for tax calculation.	Input	Optional
I_OrideRsn	25	Char		Dim(100)	Reason for applying TaxOverride	Input	Optional

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
I_TaxIncl	5	Char		Dim(100)	TaxIncluded indicates the tax amount has already been included in the line amount. The tax will be reversed out and returned in the tax property. The taxable property will indicate the calculated tax base amount.	Input	Optional
I_BINL	9	Char		Dim(100)	Optional Business Identification Number	Input	Optional

Appendix C: Tax Data Parameters

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
A Document returns the following parameters:						
O_TransID	20	Char			A unique Transaction ID identifying a specific request/response set.	Output
O_ResultSetCode	10	Char			Indicates the success (or SeverityLevel) of the operation. Success - Operation succeeded. Warning - Warnings occurred, operation succeeded. Error - Errors occurred, operation failed. Exception - Unexpected exceptions occurred, operation failed.	Output
O_DocID	10	Char			Unique ID returned by GetTax. Can be used as an input to PostTax, CommitTax, CancelTax, or AdjustTax	Output
O_DocType	25	Char			The original document's type, such as Sales Invoice or Purchase Invoice	Output
O_DocCode	20	Char			Document Code, that is, the internal reference code used by the client application.	Output
O_DocDate	10	Char			The date on the invoice, purchase order, etc.	Output

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
O_DocStatus	10	Char			<p>The document's status is returned in the GetTaxResult (except for DocStatus.Any) and indicates the state of the document in tax history.</p> <p>Temporary - A temporary document not saved (DocumentType was SalesOrder, PurchaseOrder, ReturnOrder)</p> <p>Saved - A saved document (DocumentType was SalesInvoice, PurchaseInvoice, ReturnInvoice) ready to be posted.</p> <p>Posted - A posted document (not committed). See PostTax for information on posting a document.</p> <p>Committed - A posted document that has been committed. See CommitTax for information on committing a document.</p> <p>Cancelled - A committed document that has been cancelled. See CancelTax for information on cancelling a document.</p> <p>Adjusted - A document that has been adjusted through See AdjustTax call.</p> <p>Any - Any status (used for searching)</p>	Output
O_Reconciled	5	Char			True if the document has been reconciled. Only committed documents can be reconciled.	Output
O_TimeStamp	30	Char			Date of the last status change on the document (that is, Save date, Post date, Commit date, Cancel date). This will get converted in LocalTimeZone.	Output
O_TotalAmt	15	Dec	6		The sum of all line Amount values.	Output
O_TotalDisc	15	Dec	6		The sum of all TaxLine Discount amounts. Typically it will equal the requested Discount, but, it is possible that no lines were marked as discounted.	Output
O_TotalExmpt	15	Dec	6		The sum of all TaxLine Exemption amounts.	Output
O_TotTaxable	15	Dec	6		The amount the tax is based on. This is the total of all Line Base amounts. Typically it will be equal to the document	Output
O_TotalTax	15	Dec	6		The total tax for the document.	Output

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
O_TotTaxCalc	15	Dec	6		TotalTaxCalculated indicates the total tax calculated by AvaTax. This is usually the same as the TotalTax, except when a tax override amount is specified. This is for informational purposes. The TotalTax will still be used for reporting.	Output
O_HashCode	10	Int	0		Hash Code.	Output
O_Locked	5	Char			Flag indicating if a Document has been locked by Avalara's Managed Returns Service (MRS) process. Locked documents cannot be modified and cannot be cancelled because they have been reported on Tax Return.	Output
O_AdjustRsn	3	Int	0		Returned by the AdjustTax Service. Sets a valid reason for the given AdjustTax call. Adjustment Reason is a high level classification of why an Original Document is being modified. 0 - Not Adjusted 1 - Sourcing Issue 2 - Reconciled with General Ledger 3 - Exemption Certificate Applied 4 - Price or Quantity Adjusted 5 - Item Returned 6 - Item Exchanged 7 - Bad Debt 8 - Other (Explain - Must provide AdjustmentDescription)	Output
O_AdjustDesc	50	Char			Sets description for the given AdjustTax call. Adjustment Description is required when AdjustmentReason is "Other" for enhanced tractability.	Output
O_Version	3	Int	0		Version of the web service (for example, 4.0.0.0)	Output
O_TaxDate	10	Char			Tax Date is the date used to calculate tax on the Document.	Output
O_Volatile	5	Char			Volatile Tax Rate (true or false)	Output
A Line returns the following parameters:						
O_No	10	Char		dim(100)	Line #	Output
O_TaxCod	10	Char		dim(100)	System or Custom Tax Code.	Output
O_Taxability	10	Char		dim(100)	Indicates that the taxability of the address should be checked.	Output
O_BoundLvl	10	Char		dim(100)	The level of jurisdiction boundary precision used for the tax calculation.	Output
O_Exemption	15	Dec	6	dim(100)	The portion of the detail that is exempt from taxes.	Output
O_Discount	15	Dec	6	dim(100)	The discount amount to apply to the document.	Output

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
O_Taxable	15	Dec	6	dim(100)	The taxable amount for the jurisdiction (or part of a jurisdiction in cases of tiered tax rates).	Output
O_Rate	15	Dec	6	dim(100)	The tax rate, that is the rate of taxation (0.0 - 1.0)	Output
O_Tax	15	Dec	6	dim(100)	The tax amount, that is the calculated tax (Taxable * Rate)	Output
O_TaxCalcd	15	Dec	6	dim(100)	TaxCalculated indicates the tax calculated by AvaTax. It will usually be the same as the Tax property, unless it has been overridden by a TaxAmount.	Output
O_TaxIncl	5	Char		dim(100)	TaxIncluded indicates the tax amount has already been included in the line Amount. The tax will be reversed out and returned in the Tax property. The Taxable property will indicate the calculated tax base amount.	Output
An address returns the following parameters:						
O_Address	50	Char		dim(15)	Street address	Output
O_AddrCode	3	Char		dim(15)	Programmatically determined value used internally by the adapter. Defaults to the hash code of the Address object.	Output
O_BoundLevel	3	Int		dim(15)	The level of jurisdiction boundary precision used for the tax calculation.	Output
O_City	50	Char		dim(15)	City	Output
O_Country	2	Char		dim(15)	Country code (e.g. US)	Output
O_PostCode	10	Char		dim(15)	Postal or ZIP Code.	Output
O_Region	2	Char		dim(15)	Province or state.	Output
O_TaxRegId	10	Int		dim(15)	TaxRegionId provides the ability to override the tax region assignment for an address.	Output
O_Latitude	10	Char		dim(15)	Latitude	Output
O_Longitude	10	Char		dim(15)	Longitude	Output
O_GeoCodeTyp	20	Char		dim(15)	GeoCode Type (for example, StreetLevel)	Output
O_ValidSt	20	Char		dim(15)	Validation status (for example, NormalHit)	Output
O_DistToBnd	3	Int		dim(15)	Distance to boundary.	Output
Error Messages return the following parameters:						
O_Summary	50	Char		dim(10)	Gives the concise summary of the message.	Output
O_Details	50	Char		dim(10)	Gives more specific information about the nature of the exception.	Output

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
O_HelpLink	50	Char		dim(10)	Gives the URL to help page for this message.	Output
O_RefersTo	50	Char		dim(10)	Gives the name of a particular field if the service specified one to which this exception applies.	Output
O_Severity	50	Char		dim(10)	Gives the Severity Level of the message.	Output
O_Source	50	Char		dim(10)	Gives the source of the message.	Output

Appendix D: Tax Jurisdiction Data Parameters

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
O_Country	2	Char		dim(100,10)	Country name	Output
O_RegionID	10	Char		dim(100,10)	State or Province	Output
O_JurisType	10	Char		dim(100,10)	Jurisdiction Type (for example, State or County etc.)	Output
O_JurisCode	10	Char		dim(100,10)	Jurisdiction Code for the taxing jurisdiction	Output
O_TaxType	10	Char		dim(100,10)	Tax Type S - Sales Tax U - Use Tax	Output
O_Base	15	Dec	6	dim(100,10)	The tax base, that is the taxable amount.	Output
O_Taxable	15	Dec	6	dim(100,10)	The taxable amount for the jurisdiction (or part of a jurisdiction in cases of tiered tax rates).	Output
O_Rate	15	Dec	6	dim(100,10)	The tax rate, that is the rate of taxation (0.0 - 1.0)	Output
O_Tax	15	Dec	6	dim(100,10)	The tax amount, that is the calculated tax amount (Taxable * Rate)	Output
O_TaxCalcd	15	Dec	6	dim(100,10)	TaxCalculated indicates the tax calculated by AvaTax. It will usually be the same as the Tax property, unless it has been overridden by a TaxAmount.	Output
O_NonTaxable	15	Dec	6	dim(100,10)	The portion of the detail that is not subject to taxes.	Output
O_Exemption	15	Dec	6	dim(100,10)	Exempt amount for this line.	Output
O_JurisName	25	Char		dim(100,10)	The jurisdiction name.	Output
O_TaxName	25	Char		dim(100,10)	It further defines the tax and jurisdiction.	Output
O_TaxAuthType	3	Int	0	dim(100,10)	X12+ Tax Authority Type	Output
O_TaxGroup	10	Char		dim(100,10)	TaxGroup indicates the group code when special grouping is enabled for particular client applications. For other clients, it will be empty.	Output
O_RateType	1	Char		dim(100,10)	RateType indicates the tax rate type. These currently include: F = Food (US) G = General (US) S = Standard (VAT) R = Reduced (VAT) Q = Super Reduced (VAT) A = Reduced Rate 1 (VAT) B = Reduced Rate 2 (VAT) Z = Zero Rate	Output
O_StateAssNo	4	Char		dim(100,10)	StateAssignedNo indicates the identifier given to the location by the state.	Output

Avalara Support

If you have questions about SalesTaxII, please contact Avalara Support with our toll free number (1-877) 780-4848, option 2. Standard support hours are 7:00 a.m. to 4:00 p.m. Pacific, Monday through Friday.

Requests for assistance can also be placed online via email at taxrates.support@avalara.com or via chat at <http://www.avalara.com/Technical-Support>.