# AvaTax REST V2 SDK for IBM i

## Implementation Guide

*Version 1.30*

*Revision date: 02/11/2020*

Contents

# 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 nexus, tax tiers, tax holidays, exemptions, certificate management, and product taxability rules.

AvaTax Calc is a powerful, online sales tax compliance solution. The AvaTax REST V2 SDK for IBM i (SDK) allows you to easily create an integration with AvaTax Calc. It is designed to operate with most AS400, iSeries or IBMi applications where tax calculation or address validation is required.

The SDK makes the AvaTax Calc 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 to achieve the best possible end-user experience. The SDK is fully documented, including sample code and complete certification requirements.

## Document Scope

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

# Who Should Read this Document

Any developer with an interest connecting to Avalara web services using the SDK 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 REST request, sending it the service, and parsing the returned JSON response.



Figure 1.1

# Features and Benefits of the AvaTax SDK for IBM i

The SDK:

- Is written entirely in RPG. There is no need for developers to use or understand Java. Furthermore, there is no need to implement IBM Websphere.
- Uses the latest Avalara REST v2 API interface
- Runs on all versions of the IBMi operating system from V6R1
- 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 for which a tax calculation is required.
2. The SDK adapter receives the invoice details and generates a REST enabled request which it transmits over the Internet to the Avalara tax calculation web service.
3. The SDK then receives the response message from the service, parses it, and returns the calculated tax results back to the calling application. Details of the transaction can be recorded in the customer's cloud-based AvaTax Admin Console for 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. Check that these are satisfied before beginning installation.

### Software and Hardware Requirements

- A IBMi (AS/400, iSeries, or i5) running OS/400 version V6R1 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   Installed
Program    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

## Digital Certificates

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.

For information on using Digital Certificate Manager to set up your store see Appendix E.

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 at least read/execute (*RX) authority to each of the nested directories that it is in.

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

e.g.  WRKLNK  '/'

```
                            Work with Object Links

    Directory  . . . . :

    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     /                    DIR




                                                                    Bottom
    Parameters or command
    ===>  _____
    F3=Exit    F4=Prompt    F5=Refresh    F9=Retrieve    F12=Cancel    F17=Position to
    F22=Display entire field           F23=More options
```

Then option '2' to change the authority.

```
                          Work with Authority

         Object . . . . . . . . . . . . . :    /
         Type . . . . . . . . . . . . . :    DIR
         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      *RWX         X     X     X      X
          _   QSYS         *RWX         X     X     X      X
          _   QDIRSRV      *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, 2009.
```

Continue down the chain until QIBM, USERDATA, ICSS, CERT and SERVER all have authority that looks like Fig.2.1a below. User *PUBLIC   *RX.

```
                          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





                                                                    Bottom
         Parameters or command
         ===> _____
         F3=Exit    F4=Prompt    F5=Refresh    F9=Retrieve    F12=Cancel    F17=Position to
         F22=Display entire field              F23=More options
```

```
                        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


                     Data    --Object Authorities--
Opt  User           Authority 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.
```

```
                        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


                     Data    --Object Authorities--
Opt  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 two '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




                                                         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.2.1b

Alternatively, you can use the following commands to make the necessary changes.

CHGAUT OBJ('/')  USER(*PUBLIC) DTAAUT(*RX)

CHGAUT OBJ('/QIBM') USER(*PUBLIC) DTAAUT(*RX)

CHGAUT OBJ('/QIBM/UserData') USER(*PUBLIC) DTAAUT(*RX)

CHGAUT OBJ('/QIBM/UserData/ICSS') USER(*PUBLIC) DTAAUT(*RX)

CHGAUT OBJ('/QIBM/UserData/ICSS/CERT') USER(*PUBLIC) DTAAUT(*RX)

CHGAUT OBJ('/QIBM/UserData/ICSS/CERT/SERVER') USER(*PUBLIC) DTAAUT(*RX)

CHGAUT OBJ('/QIBM/UserData/ICSS/CERT/SERVER/DEFAULT.KDB')  USER(*PUBLIC) DTAAUT(*R)

CHGAUT OBJ('/QIBM/UserData/ICSS/CERT/SERVER/DEFAULT.RDB') USER(*PUBLIC) DTAAUT(*R)

## SDK Limits

The limits for a single tax call are as follows:

- 3750 detail lines
- 500 addresses

## AvaTax Admin Console

Tax profiles as well as other company organizational settings are managed within the AvaTax Admin Console (a secure 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 or you can sign up for a free trial account by going to http://developer.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 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 Help Center (https://help.avalara.com). 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.*



Figure 2.1

## Installing the AvaTax SDK for IBM i

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

## Install the AVATAX Library onto your IBM Server

Components for SDK are contained in a single library, by default called AVATAXR2.

For **new installations**, the first step is to install the AVATAXR2 library onto your system. To make this task simple Avalara provides an installer. Running the installer will download the latest version of the SDK onto your IBMi and install it for you.

If you are **upgrading an existing installation** then you will already have the installer on your system, so simply make sure that you have AVATAXR2 (or any other Avatax SDK library) in your library list and start at step 4.0).  All your custom settings are retained in the upgraded version.

IMPORTANT Note: Always read the Change Log – Appendix H before upgrading any installation. Some upgrades may require minor changes to your own connector code before you recompile them over your newly upgraded library. If you are upgrading over multiple releases then make sure that you read and follow the Appendix H instructions for EVERY release between your current release and the latest.

**Here are the instructions for installing or upgrading the SDK.**

1.0) Sign off the IBMi

2.0) Sign on as QSECOFR or a profile with *ALLOBJ authority and make sure that QTEMP is the first library in you library list (EDTLIBL).

3.0) Download the Avatax Installer. At a command line key the following;

   3.1)    CRTSAVF FILE(QTEMP/AVASTALLER)

   3.2)    Start an FTP session by typing **ftp 'ftp.taxrates.com'** at a command line. Follow the script below.

   Connecting to host FTP.TAXRATES.COM at address 10.52.0.15 using port 21.

   220 Microsoft FTP Service

   > **ftp_avatax**

   331 Password required for ftp_avatax.

   > **Avalara!**

   230 User ftp_avatax logged in.

   Windows_NT

   > **bin**

   200 Type set to I.

   > **lcd qtemp**

   Local working directory is QTEMP

   > **cd ftp600_avatax**

   250 CWD command successful.

   > **get avastaller   (replace**

   500 'EPSV': command not understood

   227 Entering Passive Mode (10,52,0,15,19,24).

   125 Data connection already open; Transfer starting.

   226 Transfer complete.

   236544 bytes transferred in 0.043 seconds. Transfer rate 5505.024 KB/sec.

   >**quit**

3.3)　　　Restore the saved objects to QTEMP using the command below.

RSTOBJ OBJ(*ALL) SAVLIB(QTEMP) DEV(*SAVF) SAVF(QTEMP/AVASTALLER) RSTLIB(QTEMP)

4.0)　　　Run the installer by invoking the BUILDR2 command. By default BUILDR2 will install into a library called AVATAXR2. However, if you prompt the command using F4 then the installer will allow you enter any library that either does not exist or is already on your system as an Avatax SDK library. The installer will now create or update your SDK library. If you are performing an update rather than a new install, then all your custom configurations will be retained.

The installer works in 2 parts. The first part downloads the latest version of the SDK to a library called ORANGE (Note: You must have FTP enabled on your IBMi). The second part is a submitted job called RV2BUILD that updates or creates your SDK library.

At the end of the upgrade or install check the job log of the RV2BUILD job to make sure that the update was successful.



If you do not see the success message you or you receive an error message check your job log for errors. Correct the issue and try again or contact support@avalara.com for assistance.

The installer is designed to either update your library successfully or leave it alone. However, in the unlikely event that you should experience problems with your SDK library after an install a pre-update copy of your objects is saved in a library called AVABACK.

5.0) Sign off and on again to use your application.

6.0) If you have built a connector that uses the SDK then **don't forget to recompile your connector program** over the new SDK library.

# 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 menu items.

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

**Files:**

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

**Program:**

- SC0005R – User Authority Maintenance

**Command:**

- RV2 – start the menu system

To start the menu security program:

1. Add AVATAXR2 to your library list (ADDLIBLE AVATAXR2 *FIRST).
2. Type 'RV2' 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                    3/15/16
GWILSON                         Main Menu                          13:31:35


  1. Maintenance Menu                    13. INFOR XA Setup Options
  2. Inquiry Menu




                                         20. Integration Tester Defaults
                                         21. Integration Tester

 11. Administration Menu                 23. Work with Spool Files
                                         24. Signoff


Enter Option: __



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

Figure 3.1

4. Select option 11 – Administration Menu

5. 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

# Configuring the System Control File

All information needed to connect and authenticate to the Avalara web services on the development server is provided with the SDK and contained in the system control file (RV2001P. You just need to add your company and the Account and Password provided by Avalara. When you decide to go live with your application you need to make sure you create a system control file entry that has your production company and credentials with Account Type = 'P'.

```
RV2001R                Edit System Control  File Record            3/21/19
                                                                   13:03:32
   Company: 1_____      Active: Y

Operation: create
           (create, adjust, commit, void , resolve, transactions)
Authorization String (BASE64 encoded):
 Basic R3JhjsjDGb5zdzk4MTEwIQ==_____

     Account Type: S (S-andbox or P-roduction)

Request Filename: _____   (Opt)
 Result Filename: /tmp/*Auto_____   (Opt)

Compact JSON Y/N: N          Avatax Request Time Out (secs): ___30

                             Return Basic Tax Details Only: N

Debug Y/N: Y Debug Filename: /tmp/*auto_____
                             (e.g /tmp/mydebug.txt)



F3=Exit F12=Cancel

MA    A            MW                                        19/036
```

Figure 2.2

- **Company**

  Set this to the same value as you used for company in your admin console.


- **Active**

  Used to switch the service on or off.


- **Operation**

  This is the operation that you wish to perform. For example, the 'create' operation allows you to calculate taxes and optionally record the result in your admin console.

  See https://sandbox-rest.avatax.com/swagger/ui/index.html#/  or https://developer.avalara.com/ for more details regarding operations and what they do.

- **Authorization String**

This should be entered as a BASE64 encoded string version of your Avalara ID and password.

You can obtain the BASE64 encoded string by using an online encoder like the one at …

http://www.webtoolkitonline.com/base64-converter.html

Enter your Avalara provided user Id and password separated by a colon and then hit the encode button. Your credentials will be converted into an encoded string that can be cut and pasted into your system control file record.
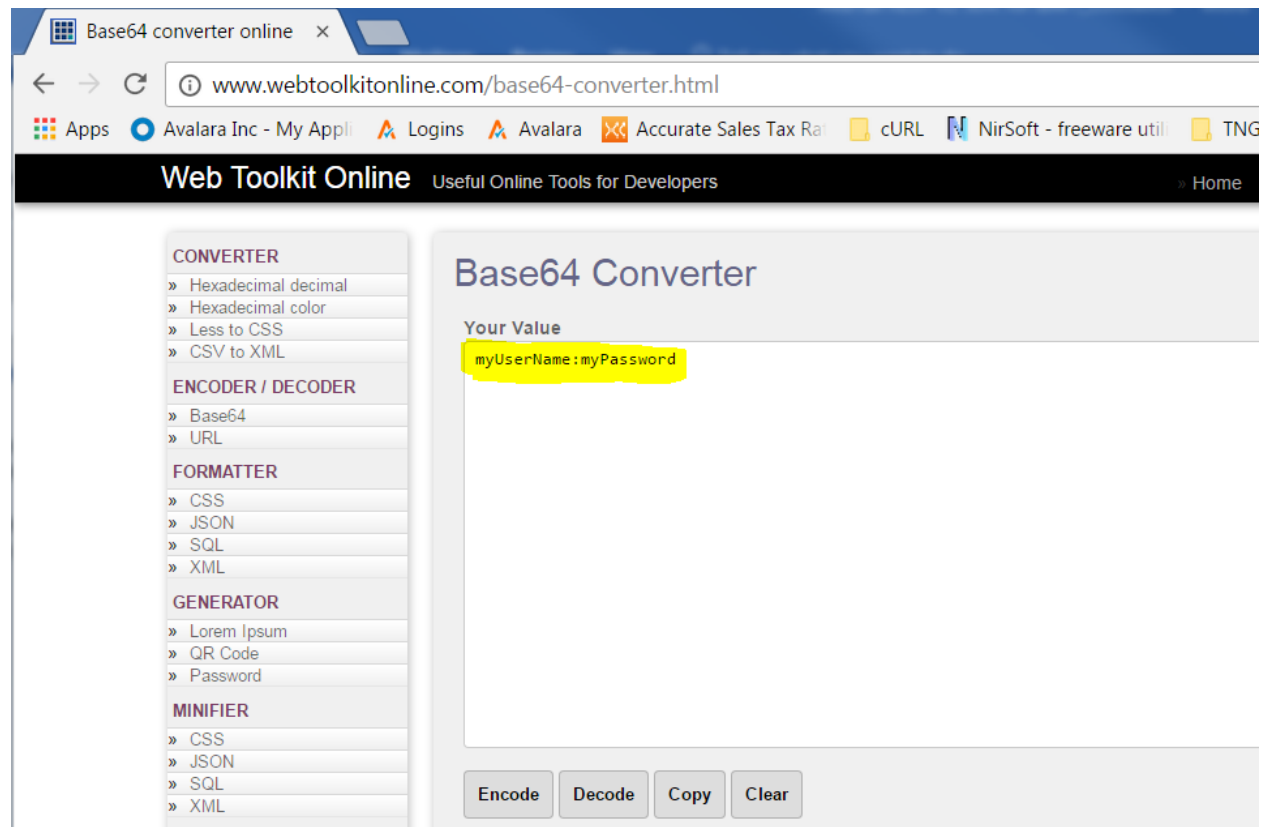


- **Account Type**

  Select 'S' to use a development (sandbox) account.

  Select 'P' to use a production account.

- **Request Filename**
  You can optionally enter an IFS location and file that will be used to build and stream your outgoing web service message. If you leave it blank the request will be handled in main memory which will give slightly faster performance – **this is the recommended method.** However, if you need access to the outgoing JSON data stream then you can use the file approach. From release V1.21 you may use the *Auto feature. Entering *Auto in this field will force the SDK adapter to create a unique file in the IFS for every Avatax request, using the format reqnnnnnnn.txt where 'n' is a unique integer (e.g. req0000004.txt). Files are recycled when the integer gets above 9999999. The default IFS directory is /tmp. So entering just *Auto will create the files in /tmp. If you wish to use another directory then specify it as follows /myDirectory/*Auto. The directory myDirectory must exist on the IBMi, the SDK does not create new directories.

- **Result Filename**
  You can optionally enter an IFS location and file that will be used to receive your incoming result stream. If you leave it blank the request will be handled using a default file. you may use the *Auto feature. Entering *Auto in this field will force the SDK adapter to create a unique file in the IFS for every response in the format respnnnnnnn.txt where 'n' is a unique integer (e.g. resp0000114.txt). Files are recycled when the integer gets above 9999999. The default IFS directory is /tmp. So entering just *Auto will create the files in /tmp. If you wish to use another directory then specify it as follows /myDirectory/*Auto. The directory myDirectory must exist on the IBMi, the SDK does not create new directories.

- **Compact JSON**
  Enter Y to compact the outgoing JSON request for more efficient performance.

- **Avatax Request Time Out**
  You can adjust the time (in seconds) that the SDK adapter waits for a response. If you experiencing HTTPS time out issues then increase this value.

- **Return Basic Tax Details Only**
  This Y or N flag gives the customer the ability to have the SDK adapter return basic tax details only. This improves performance and reliability for those customers who only require access to the following header data.

| | |
|---|---|
| taxDate | The date on which this transaction occurred |
| totAmount | The total amount of this transaction |
| totExempt | The amount of this transaction that was exempt |
| totTax | The total tax calculated for all lines in this transaction |
| totTaxable | The portion of the total amount of this transaction that was taxable |
| totTaxCalcd | If a tax override was applied to this transaction, indicates the amount of tax Avalara calculated for the transaction |

- **Debug Y/N**

  Entering Y in this field will put the SDK into debug mode and a comprehensive report will be produced for all transactions.

- **Debug Filename**

  You can optionally enter the IFS directory and filename that you would like to record the debug data to. Leaving it blank will result in an output file called httpapi_debug.txt.

  **Note:** Debug files are overwritten by subsequent transactions. This can be frustrating in a busy environment where you are trying to get results for a specific document. To overcome this, use *Auto for the filename. This will force the SDK to generate a new filename for each call in the format avatax_debugnnn.txt where nnn is a unique number between 001 – 999. When the number hits 999 the files will start to recycle from 001 again.

  e.g. /tmp/*Auto   will create a unique file in the IFS tmp directory.

# 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>.

```
RV2001R              Maintain Avalara System Control File              5/19/17
                                                                      12:55:43
Type option, press Enter.
 2=Edit   3=Copy   4=Delete   5=Display   T=Test Credentials


Opt Company Code              Operation            Environment   Active
 _   1                        adjust               Sandbox          Y
 _   1                        commit               Sandbox          Y
 T   1                        create               Sandbox          Y
 _   1                        resolve              Sandbox          Y
 _   1                        transactions         Sandbox          Y
 _   1                        void                 Sandbox          Y
 _   2                        adjust               Sandbox          Y
 _   2                        commit               Sandbox          Y
 _   2                        create               Production       Y




F1=Info    F3=Exit    F6=Add
Copyright 2004-2016 Avalara, Inc. All Rights Reserved
```

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

```
RV2001R               Maintain Avalara System Control File              5/19/17
                                                                        12:55:43
 Type .........................................................................
  2=E :                   Test Connection Settings Result                      :
      :                                                                        :
 Opt  : Result: Congratulations you have connected to Avatax                   :
  _   :                                                                        :
  _   :                                                                        :
  T   : Detail: Version: 17.5.1-73                                             :
  _   :         Authenticated: true                                           :
  _   :         Auth Type: UsernamePassword                                    :
  _   :         User Name: Graham                                              :
  _   :         User Id: 13706                                                 :
  _   :         User Acct: 1100057680                                          :
  _   :                                                                        :
      :    Exit=<Enter>                                                        :
      :                                                                        :
      :........................................................................:




 F1=Info    F3=Exit    F6=Add
 Copyright 2004-2016 Avalara, Inc. All Rights Reserved
```

Figure 3.7

3. If you get the message shown in Figure 3.8 then re-check your **Authentication String** and **Digital Certificate** set up details. These are the two most common forms of failure. Firewalls are another common issue. Make sure that you have port 443 open and that you can successfully ping the following two URI's from your IBMi command line.

ping 'sandbox-rest.avatax.com'

ping 'rest.avatax.com'

```
 RV2001R              Maintain Avalara System Control File            5/19/17
                                                                     13:14:27
 Type ......................................................................
  2=E :                      Test Connection Settings Result             :
      :                                                                  :
 Opt  : Result: Failed to connect to Avatax                              :
  _   :                                                                  :
  _   :                                                                  :
  _   : Detail: 1) Check your authentication string                      :
  _   :         2) Check your digital certificate store set up           :
  _   :            and authorization                                     :
  _   :         3) If checks 1 and 2 are OK then put the transaction     :
  _   :            that you are testing in debug mode, re-run the test   :
  _   :            and check the log for errors.                         :
  T   :                                                                  :
      :    Exit=<Enter>                                                  :
      :                                                                  :
      :..................................................................:




 F1=Info   F3=Exit   F6=Add
 Copyright 2004-2016 Avalara, Inc. All Rights Reserved
```

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 you have completed installation and configuration and 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 many of the transaction scenarios allowed by the AvaTax web service.

```
RV2003R                    REST V2 Integration Tester                    5/19/17
  Dft: 01   Company: 1_____         Operation: create_____
        Ship TO:                           Ship FROM:
  Line1: 100 Ravine Lane_____         _____
  Line2: _____       _____
   City: Bainbridge Island_____           Seattle_____
  State: WA Zip: 98110_____   Country: US   WA   98101_____              US
Lat/Lng: _____.000000   _____.000000        _____.000000   _____.000000


  DocDate: 2017-05-19 DocCode: _____ EntUse: _   Record: N Commit: N
   CustNo: Cus100 PO#: 123456__  Exempt Code: _____
  O/ride=> Amt: _____.00   Typ: _____ Date: 2017-03-28 Rsn: _____
 Adjust=> Rsn: _____    Desc: _____


  #  Item Code/Desc   Inc Ent Tax Code Qty   Amount   Rate%      Tax     Total
  1  Lumber 2 x 6_____   N  _ _____ __1  _100.00_  9.000     9.00    109.00
     2 x 6 pressure treated_____
  2  CAT PS2-10_____    N  _ _____ __2  __80.00_  9.000     7.20     87.20
     7/16 CAT PS2-10 OSB Sheathing___
  3  Freight_____    N  _ FR020100 _1  __75.00_  9.000     6.75     81.75
     Freight charge_____
                                             --------  ----------
                                               22.95     277.95

  F1=Help  F3=Exit  F4=Refresh
```

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.

```
 RV2004R                  Maintain Test Defaults - Select              5/19/17
                                                                       16:08:50

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


 Opt  Key Description
   █    FL  Create - No record
   _    LL  Use Lat/Long
   _    S1  NY Example - Using Address
   _    01  Create - No record
   _    02  Void a Transaction
   _    03  Commit a Transaction
   _    04  Adjust a Transaction
   _    05  Retrieve a Transaction












 F1=Help  F3=Exit  F6=Add
```

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

```
RV2004R                   REST V2 Integration Tester Defaults              5/19/17
  Dft: 01  Create - No record _____                     16:10:07


   Co: 1_____     Transaction Type: create_____


        Ship TO:                              Ship FROM:
   Line1: 100 Ravine Lane_____       _____
   Line2: _____      _____
    City: Bainbridge Island_____          Seattle_____
   State: WA Zip: 98110_____   Country: US    WA   98101_____              US
 Lat/Lng: ____.000000 ____.000000             ____.000000 ____.000000


  DocDate: 0001-01-01 DocCode: _____  EntUse: _  Record: N Commit: N
   CustNo: Cus100 PO#: 123456__  Exempt Code: _____
  O/ride=> Amt: _____.00_ Type: _____  Date: 2017-03-28 Rsn: _____
 Adjust=> Rsn: _____     Desc: _____


   #  Item Code        Description                Inc Ent Tax Code Qty Amount
   1  Piano_____ Bechstein concert piano_____ N   _ _____  1  5000.00
   2  Violin_____ Stradivarius demo model_____ N   _ _____  2   800.00
   3  Oboe_____ Ordinary oboe_____ N   _ _____  1   550.00



  F1=Help  F3=Exit  F6=Add
```

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.

# Canadian Taxes

Some users will have the need to print a summary of the Canadian taxes on their invoices or documentation. The Avatax API returns this data in detail form per order line but does not provide a summary for the entire document. To make summarizing Canadian taxes easier Avalara provides a program called RV2009R which can be used to return a summary of Canadian taxes for a given invoice. This program is included in your AVATAXR2 library. A demo program showing how to use RV2009R is included as menu option 2 on the INQUIRY menu. The source is in AVATAXR2/QRPGLESRC RV2009RT.

```
5761WDS V7R1M0  100416              SEU SOURCE LISTING                11/03/17 10:41:12   TAX1        PAGE    1
SOURCE FILE . . . . . . .   RV2DEV/QRPGLESRC
MEMBER  . . . . . . . . .   RV2CANTAX
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100      // Incude this copy member                                                                  11/03/17
  200      /copy RV209CP                            CA Tax Parameters                                  04/10/17
  300                                                                                                  02/22/17
  400      /free                                                                                       11/03/17
  500        // Clear TAX Buffer                                                                        02/23/17
  600        clear TAX;                                                                                 02/23/17
  700        // Set the Company and Document (Order/Invoice #) Codes                                    11/03/17
  800        TAX.Company = Company;                                                                     11/03/17
  900        TAX.DocCode = DocCode;                                                                     11/03/17
 1000        // Get the Canadian Tax Breakdown                                                          11/03/17
 1100        GetCanTax(TAX);                                                                            11/03/17
 1200        // Check the error code is *Blank and extract the results                                  11/03/17
 1300        if TAX.ErrCode = *Blanks;                                                                  11/03/17
 1400          // TAX.GstTaxName contains the GST Tax Name e.g. CANADA GST/TPS                          11/03/17
 1500          // TAX.GstRate contains the GST Tax % Rate e.g. 5.00                                     11/03/17
 1600          // TAX.GstTax contains the GST Tax Amount e.g. 30.00                                     11/03/17
 1700          // TAX.PstTaxName contains the PST Tax Name e.g. ONTARIO HST                             11/03/17
 1800          // TAX.PstRate contains the PST Tax % Rate e.g. 8.00                                     11/03/17
 1900          // TAX.PstTax contains the PST Tax Amount e.g. 48.00                                     11/03/17
 2000        else;                                                                                      11/03/17
 2100          // Handle error condition                                                                11/03/17
 2200          // TAX.ErrCode contains 'Error'                                                          11/03/17
 2300          // TAX.ErrText contains the error message                                                11/03/17
 2400        endif;                                                                                     11/03/17
 2500      /end-free                                                                                    02/21/17
                          * * * *  E N D  O F  S O U R C E  * * * *
```

**Note**: RV2009R uses the Avalara **transactions** operation, so make sure that you activate that function with your company code and valid credentials in your System Control file.

# Developing Your Connector

This section gives guidance and examples of how to use the SDK to connect to Avalara web services. If you remember from Figure 1.1, the SDK 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 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 following is an explanation of the data structures used by the SDK.

### SysCtl - HTTP Parameters Used by the Web Service Call

These parameters are easy to populate. Simply chain to the System Control file using a key of Company Code and Operation type (e.g. ctl_cmpy = '1'  ctl_oper='create') and then populate the SysCtl data structure with the returned results.

Refer to Appendix A for a table of SysCtl parameters, lengths, and types.

### TaxHdr – Document Header Details

This structure is used to pass the order or invoice header details and also to receive document level taxation details like the Total Tax or Total Amount for a document.

Refer to Appendix A for a table of parameters, lengths, and types.

### TaxLin – Document Line Details

This structure is used to pass the order or invoice line details.

Refer to Appendix A for a table of parameters, lengths, and types.

### TaxDet – Tax Line Details Returned by AvaTax

This structure 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.).

## TaxAdr – Addresses used for Calculation Returned by AvaTax

This structure is returns the actual addresses and boundary level information used by Avatax when calculating taxes.

Refer to Appendix A for a table of parameters, lengths, and types.

## TaxSum – Summary of Taxes Returned by AvaTax

This structure returns a summary of the of the taxes calculated by Avatax and separated by actual taxing jurisdiction. So, for example, if you were delivering to an address in Los Angeles, CA then this data structure would contain the document level summary totals for each individual taxing jurisdiction, as shown in the example below.

| Tax Name | | Rate | Amount |
|---|---|---|---|
| CA STATE TAX | CALIFORNIA | 6.000 | 60.000 |
| CA COUNTY TAX | LOS ANGELES | .250 | 2.500 |
| CA SPECIAL TAX | LOS ANGELES COUNTY | 1.500 | 15.000 |
| CA SPECIAL TAX | LOS ANGELES CO LOCAL | 1.000 | 10.000 |

Refer to Appendix A for a table of parameters, lengths, and types.

## TaxMsg – Returns Error Messages

This structure returns error messages sent by the service.

Refer to Appendix A for a table of parameters, lengths, and types.

## Coding Considerations

In your AVATAXR2 library you will find a source member called QRPGLESRC/RV2EXAMPLE. This is not a working program but it will take you through all the steps required to complete your AVATAX connector using the SDK. Just follow the steps 1 through 9.

```
Columns . . . :    6 100                         Browse                              AVATAXR2/QRPGLESRC
SEU==> _____                                          RV2EXAMPLE
FMT ** ... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
        *************** Beginning of data ***************************************************************
0001.00 // *****************************************************                    150827
0002.00 // * STEP 1 - include the following H-Specs                                 150827
0003.00 // *****************************************************                    150827
0004.00 //                                                                          150827
0005.00 H DFTACTGRP(*NO) BNDDIR('HTTPAPI')                                           150803
0006.00 H BNDDIR('YAJL') DECEDIT('0.')                                              150803
0007.00 H OPTION(*SrcStmt:*NoDebugIO)                                               150803
0008.00 //                                                                          150827
0009.00 // *****************************************************                    150827
0010.00 // * STEP 2 - include filespec for the System Control file                  150827
0011.00 // *****************************************************                    150827
0012.00 //                                                                          150827
0013.00 F* RV2 - System Control File                                                170414
0014.00 FRV2001P   if   e         k disk                                            170414
0015.00 //                                                                          150827
0016.00 // *****************************************************                    150827
0017.00 // * STEP 3 - include the following copy members and                        170414
0018.00 // *         define the System Control key list                             170414
0019.00 // *****************************************************                    150827


 F3=Exit   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle   F12=Cancel
 F16=Repeat find       F24=More keys
                              (C) COPYRIGHT IBM CORP. 1981, 2007.
```

Also look at the source for some of the other demo programs like the Integration Tester (RV2003R) to see how they were used to return a tax calculation.

# Easy Operations – Canada Tax, Void Tax, Commit Tax, Refund Tax and Retrieve Document

For your convenience, the SDK includes some pre-programmed functions that make it easy to retrieve the **GST/PST breakdown** of a Canadian invoice, **Void** a tax transaction, **Commit** a previously uncommitted transaction, **Refund** a transaction or **Retrieve** the details of a previously calculated document. This section describes how to use them.

## Retrieve Canadian Tax Breakdown

This function will retrieve the GST/PST breakdown for a Canadian. Useful if you need to print those details on an invoice. The source code showing how to call this function can be found in AVATAXR2/QRPGLESRC RV2CANTAX.

```
SOURCE FILE . . . . . . .   AVATAXR2/QRPGLESRC
MEMBER   . . . . . . . . .   RV2CANTAX
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100        // Incude this copy member
  200        /copy RV209CP                         CA Tax Parameters
  300
  400        /free
  500           // Clear TAX Buffer
  600           clear TAX;
  700
  800           // Set the Company and Document (Order/Invoice #) Codes
  900           TAX.Company = myCompany;
 1000           TAX.DocCode = myDocCode;
 1100
 1200           // Get the Canadian Tax Breakdown
 1300           GetCanTax(TAX);
 1400
 1500           // Check the error code is *Blank and extract the results
 1600           if TAX.ErrCode = *Blanks;
 1700              // TAX.GstTaxName contains the GST Tax Name e.g. CANADA GST/TPS
 1800              // TAX.GstRate contains the GST Tax % Rate e.g. 5.00
 1900              // TAX.GstTax contains the GST Tax Amount e.g. 30.00
 2000              // TAX.PstTaxName contains the PST Tax Name e.g. ONTARIO HST
 2100              // TAX.PstRate contains the PST Tax % Rate e.g. 8.00
 2200              // TAX.PstTax contains the PST Tax Amount e.g. 48.00
 2300           else;
 2400              // Handle error condition
 2500              // TAX.ErrCode contains 'Error'
 2600              // TAX.ErrText contains the error message
 2700           endif;
 2800        /end-free
                         * * * *  E N D  O F  S O U R C E  * * * *
```

## Void Tax

This function allows you to Void an existing document by passing your company code, document code and document type. The source code showing how to call this function can be found in AVATAXR2/QRPGLESRC RV2VOID.

```
SOURCE FILE . . . . . . .   RV2DEV/QRPGLESRC
MEMBER   . . . . . . . . .   RV2VOID
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100        // Incude this copy member
  200        /copy RV211CP                         Void Tax Parameters
  300
  400           // Clear TAX Buffer
  500           clear VTAX;
  600
  700           // Set the details of the document to be voided
  800           VTAX.Company = myCompany;  // e.g. '1'
  900           VTAX.DocCode = myDocCode;  // e.g. '0123456'
 1000           VTAX.DocType = myDocType;  // e.g. 'SalesInvoice'
 1100
 1200             // Void the document
 1300             VoidTax(VTAX);
 1400
 1500           // Check the Result
 1600           if VTAX.RtnCode <> 'Success';
 1700              // Handle error condition
 1800              // VTAX.RtnCode contains 'Error'
 1900              // VTAX.RtnText contains the error message
 2000           endif;
 2100        /end-free
                         * * * *  E N D  O F  S O U R C E  * * * *
```

## Commit Tax

This function allows you to Commit an existing document by passing your company code and document code. The source code showing how to call this function can be found in AVATAXR2/QRPGLESRC RV2COMMIT.

```
SOURCE FILE . . . . . . .   AVATAXR2/QRPGLESRC
MEMBER   . . . . . . . . .   RV2COMMIT
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
   100        /copy RV212CP                              Commit Tax Parameters
   200
   300        /free
   400           // Clear TAX Buffer
   500           clear CTAX;
   600
   700           // Get the invoice to Commit from the screen input
   800           CTAX.Company = Company;
   900           CTAX.DocCode = DocCode;
  1000
  1100           // Commit the document
  1200           CommitTax(CTAX);
  1300
  1400           if CTAX.RtnCode <> 'Success';
  1500             // Handle error condition
  1600             // VTAX.RtnCode contains 'Error'
  1700             // VTAX.RtnText contains the error message
  1800           endif;
  1900
  2000           result = CTAX.RtnCode;
  2100           errsum = CTAX.RtnText;
  2200        /end-free
                      * * * * E N D  O F  S O U R C E  * * * *
```

## Refund Tax

This function allows you to quickly and easily create a ReturnInvoice representing a full or partial refund for a previously created SalesInvoice transaction.

The program RV2014RT is a working example of this function and is available from the SDK Inquiry menu.  The source member AVATAXR2/QRPGLESRC RV2REFUND contains the code needed to use this function which you can copy into your own program and modify accordingly.

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
   100      H OPTION(*SECLVL:*SRCSTMT:*NODEBUGIO)
   200      H DFTACTGRP(*NO) ACTGRP('AVATAXR2') BNDDIR('AHTTPAPI':'QC2LE')
   300
   400       /copy RV214CP                              Refund Tax Parameters
   500
   600       /free
   700          // Clear TAX Buffer
   800          clear RTAX;
   900
  1000          // Set the details of the document to be refunded
  1100            RTAX.Company = myCompany;        // e.g. '1'
  1200
  1300            RTAX.DocCode = myDocCode;        // The document code to refund
  1400
  1500            RTAX.NewDocCode = myNewDocCode; // The credit memo document # to
  1600                                            // be created. If left blank then
  1700                                            // myDocCode + '_CR' will be used.
  1800
  1900            RTAX.DocType = DocType;          // The document type being credited
  2000                                            // e.g. SalesInvoice (optional defaults
  2100                                            // to SalesInvoice)
  2200
  2300            RTAX.Date = %char(%date());      // The date applied to to the credit
  2400
  2500            RTAX.Reason = Reason;            // The reason for the credit
  2600
  2700            RTAX.RefundDteOvr = 'false';     // The refund date override 'true' or 'false' (Opt)
  2800
  2900            RTAX.RefundType = RefundType;    // The refund type ('Full', 'Partial', 'TaxOnly'
  3000                                            // 'Percentage')
  3100
  3200            RTAX.RefundPct = RefundPct;      // The refund percent e.g. 50 (optional)
  3300
  3400            RTAX.RefundLines(1) = '001';   // The lines to refund (optional)
  3500            RTAX.RefundLines(2) = '003';   // The lines to refund (optional)
  3600
  3700            // Refund the document
  3800            RefundTax(RTAX);
  3900
  4000          // Check the Result
  4100          if RTAX.RtnCode <> 'Success';
  4200            // Handle error condition
  4300            // RTAX.RtnCode contains 'Error'
  4400            // RTAX.RtnText contains the error message
  4500          endif;
  4600
  4700            *inlr = *on;
  4800       /end-free
```

## Retrieve Document Details

This function retrieves the details of a previously calculated document. The source code showing how to call this function can be found in AVATAXR2/QRPGLESRC RV2RTV. The results are returned in the same data structure and format as used by 'create'. If the call was successful, then SysCtl.errorCode will contain 'Success' and SysCtl.errorText will contain a successful completion message (e.g. 0123456 retrieved OK.) If there was a problem, then SysCtl.errorCode will contain the error code returned by the service and SysCtl.errorText will contain the error description (e.g. Document not found).

The program RV2013RT is a working example of this function. It can be called as follows:

CALL PGM(RV2013RT) PARM('myCompany' 'myDocumentCode')

```
100       H OPTION(*SECLVL:*SRCSTMT:*NODEBUGIO)
200       H DFTACTGRP(*NO) BNDDIR('HTTPAPI':'QC2LE')
300
400         // Include the Retrieve Tax Prototype
500        /copy RV213CP
600
700        /free
800
900         // Clear the tax buffers
1000        clear TaxHdr;
1100        clear TaxLin;
1200        clear TaxDet;
1300        clear TaxAdr;
1400        clear TaxSum;
1500        clear TaxMsg;
1600
1700        // Set the Company Code and the Document Code to retrieve
1800        TaxHdr.companyCode = myCompany;   // e.g. '1'
1900        TaxHdr.code = myDocCode;          // e.g. '0123456'
2000
2100        // Retrieve the document
2200          RetrieveTax(SysCtl:
2300                      TaxHdr:
2400                      TaxLin:
2500                      TaxDet:
2600                      TaxAdr:
2700                      TaxSum:
2800                      TaxMsg);
2900        /end-free
                      * * * *  E N D  O F  S O U R C E  * * * *
```

# 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 'create' 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 |

2. Set the DocType to ReturnInvoice (very important if you are re-using the original DocCode).
3. Set the DocDate to the tax reporting month that you want the return to appear in (typically the current month).
4. Set the TaxOverride property to TaxDate.
5. Set the TaxOverrideDate to the date the original invoice computed tax. AvaTax calculates the tax based on this date.
6. Set the AdjustmentReason to Return Items (or whatever makes sense for your business needs).
7. 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.*

8. 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 a document. Avalara has a number of pre-defined freight codes to cover various business scenarios but a good default code to use is FR020100 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 create operation should pass 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.



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 'create' 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.

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

- errorCode = SeverityLevel.**Success**
- errorCode = SeverityLevel.**Error**
- errorCode = SeverityLevel.**Exception**

*NOTES*
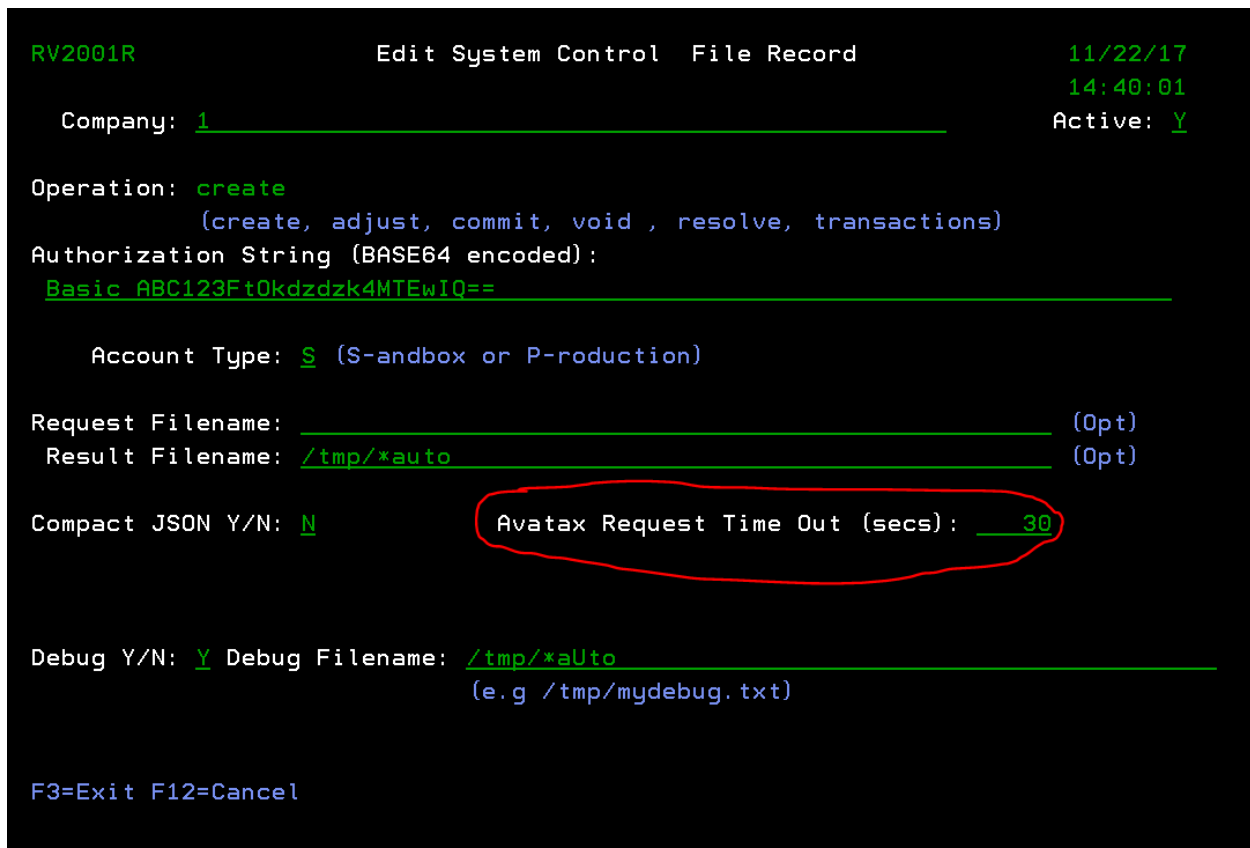
- *Typically,* **Exception** *is only returned by a system error from the Avalara web service and will be caused by circumstances outside of the client application's control (that is not having an Internet connection will return the error).*
- *In general, the path taken for errorCode=* **Exception** *should be the same as for errorCode=***Error**.

# Time Outs

Although HTTP communications are generally very fast and reliable, occasionally you may experience a time out error. This error happens when the SDK adapter is waiting for a response but does not receive one within the specified time out interval, so it gives up waiting. This could be due to several different factors. For example, maybe someone is rebooting a router or firewall, disconnecting from the internet temporarily for maintenance or maybe your ISP has an issue.

Here are some things you can do to reduce the risk of time outs.

1) In the System Control file you can configure the time out interval in seconds. If you are experiencing time outs, try increasing the wait time.

```
RV2001R                 Edit System Control  File Record            11/22/17
                                                                    14:40:01
   Company: 1_____      Active: Y

Operation: create
           (create, adjust, commit, void , resolve, transactions)
Authorization String (BASE64 encoded):
 Basic ABC123FtOkdzdzk4MTEwIQ==_____

    Account Type: S (S-andbox or P-roduction)

Request Filename: _____  (Opt)
 Result Filename: /tmp/*auto_____   (Opt)

Compact JSON Y/N: N         Avatax Request Time Out (secs):  ___30



Debug Y/N: Y Debug Filename: /tmp/*aUto_____
                             (e.g /tmp/mydebug.txt)



F3=Exit F12=Cancel
```

2) Make sure that you are on the latest release of the SDK. The SDK adapter from release 1.7 onwards has some built in resilience. It traps time out errors and resubmits the request up to 5 times. If after 5 attempts a successful response is not received, then it returns an error to the caller with SysCtl.errorCode = 'Error' and SysCtl.errorText = 'CommSSL_Read: Timeout'.

From release v1.8 the time out resilience was improved to deal with the situation where a transaction was successfully recorded and committed in Avatax but due to a time out error the details were not returned to the caller. In this case, subsequent attempts to commit the transaction results in a '**DocStatus is invalid for this operation.**' error. The SDK is now able to automatically detect this situation and will attempt to retrieve the missing details and return them to the caller who will see it as a normal, successful

transaction. After 5 attempts, if the adapter continues to receive time out errors it will return an error to the caller with SysCtl.errorCode = 'Error' and SysCtl.errorText = 'CommSSL_Read: Timeout'.

From v1.8 a feature has been included that allows a developer to force a time out situation. This is useful for testing time out error handling functions. It is recommended that developers use this feature with a 'SalesOrder' transaction type as 'SalesInvoice' transactions are subject to the time out resilience feature described above and may not return a 'CommSSL_Read: Timeout' error. To enable the Force Timeout feature set the data area AVATAX2/RV2008DA = 1.

**Note**: Do not forget to set the data area back to 0 once testing is complete.


Turn Force Timeout ON ….

CHGDTAARA DTAARA(AVATAXR2/RV2008DA) VALUE(1)

Turn Force Timeout OFF ….

CHGDTAARA DTAARA(AVATAXR2/RV2008DA) VALUE(0)

# Address Validation - 'create' 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 **'create'** request to be called devoid of a valid street address or even ZIP Code, the results may 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 **'create'** call attempts to normalize an address. 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 **'create'** operations will return incorrect results.

# Understanding Boundary levels

Boundary levels, returned by Avatax, 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.*

# Useful Links

## Web Service URLs

These are the development and production web service URLs.

### Development

https://sandbox-rest.avatax.com

### Production

https://rest.avatax.com

## developer.avalara.com

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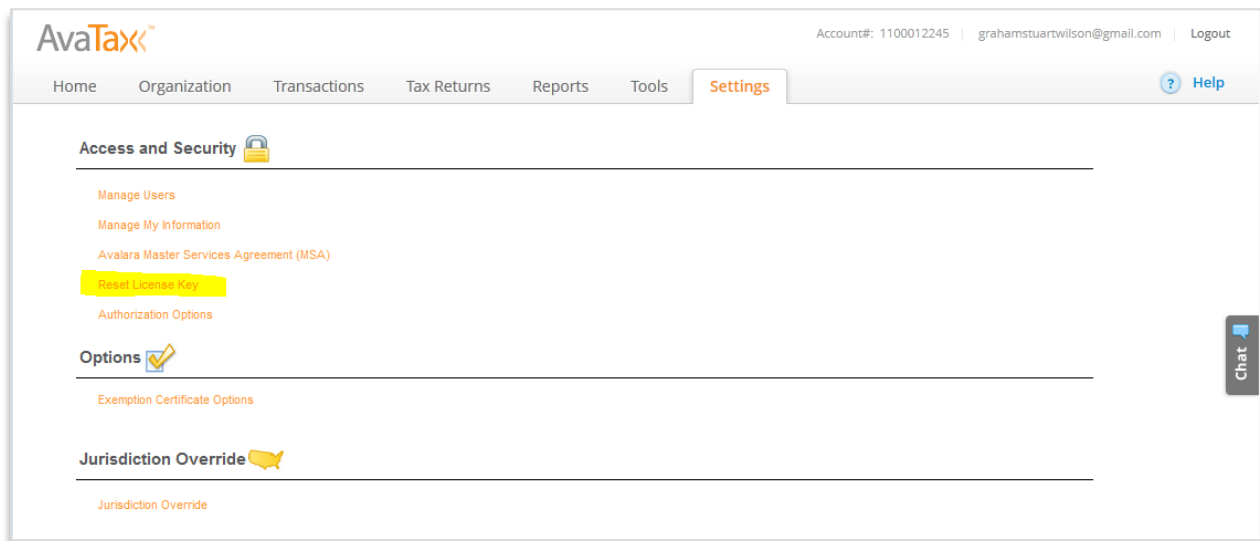


## NOTE

- *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 tax calculation may include:

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

# 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 business activity being 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. 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.

# Item Code versus Tax Code

Item Code versus Tax Code in line Items:

- Special product taxability rules can be defined using Item and/or Tax Codes:

    - Line Item Code or Tax Code

- Item Codes represent individual products, whereas Tax Codes represent categories of products. For example:

    - Item Codes 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 Tax Codes.

**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 Item Codes to a Tax Code(s) then send the Item Code for your individual product SKU.

    a. Using the example above, Item Codes M, E, and C would be mapped to each via TaxCode GR. Then if Line.Item Code = M, E or C they would automatically map to Tax Code GR, and the rule for Tax Code GR will apply.

2. Send the Tax Code GR in Line Tax Code 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 Tax Code(s) for them.

If you send a value in *both* the Item Code and the Tax Code fields, the behavior is as follows:

1. If the **Item Code** sent maps to a **Tax Code**, the mapped Tax Code will be used for product taxability rules, regardless of what is sent in the Tax Code field.
2. If the **Item Code** sent does not map to a **Tax Code**, the Tax Code sent will be used for product taxability rules.
3. Lines that have, or are mapped to **Tax Codes** 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 **void**).
- 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.

# How to Go Live

Eventually you will want to point your transactions to the Avalara production server and start calculating and recording real data.

To do this you need to set up an entry in the System Control file that points to the production rather than the development server.

You will need to change the Company, Authorization String and Account Type settings. Company and Authorization String must be your production credentials supplied to you by Avalara. Also, change The account type to be 'P' which will point your transactions to the Production URL.

```
RV2001R                    Edit System Control  File Record              6/05/17
                                                                        15:13:16
      Company: 2                                              Active: Y

   Operation: create
              (create, adjust, commit, void , resolve, transactions)
   Authorization String (BASE64 encoded):
   R3JhaGFtDfr44FRfMTEwIQ==

       Account Type: P (S-andbox or P-roduction)

   Request Filename: _____  (Opt)
    Result Filename: _____  (Opt)

   Compact JSON Y/N: Y           Avatax Request Time Out (secs):    10



   Debug Y/N: Y Debug Filename: /tmp/graham.txt
                                (e.g /tmp/mydebug.txt)



   F3=Exit F12=Cancel
```

If your test and production systems are on the same IBMi server then you can either modify the setting in entry you have been using for testing or you can use option 3 to copy a service (to a different company code) and set that to point to the production server. This way you can have both production and test applications residing on the same box. See example for the 'create' service below which has a test company under company 1 and a production company under company 2.

```
RV2001R              Maintain Avalara System Control File              6/05/17
                                                                      15:19:28
Type option, press Enter.
 2=Edit  3=Copy  4=Delete  5=Display  T=Test Credentials


Opt Company Code            Operation            Environment    Active
 _   1                      adjust               Sandbox        Y
 _   1                      commit               Sandbox        Y
 _   1                      create               Sandbox        Y
 _   1                      resolve              Sandbox        Y
 _   1                      transactions         Sandbox        Y
 _   1                      void                 Sandbox        Y
 _   2                      adjust               Sandbox        Y
 _   2                      commit               Sandbox        Y
 _   2                      create               Production     Y




F1=Info    F3=Exit    F6=Add
Copyright 2004-2016 Avalara, Inc. All Rights Reserved
```

# Appendix A: SDK Parameters

| Field Name | Size/Type | Input or Output | Description |
|---|---|---|---|
| **Tax Document Header** | | | |
| **TaxHdr** | | | |
| type | 25a | Input | Document Type: if not specified, a document with type of SalesOrder will be created by default = ['SalesOrder', 'SalesInvoice', 'PurchaseOrder', 'PurchaseInvoice', 'ReturnOrder', 'ReturnInvoice', 'InventoryTransferOrder', 'InventoryTransferInvoice', 'ReverseChargeOrder', 'ReverseChargeInvoice', 'Any'], |
| code | 50a | Input | Transaction Code - the internal reference code used by the client application. This is used for operations such as Get, Adjust, Settle, and Void. If you leave the transaction code blank, a GUID will be assigned to each transaction. |
| companyCode | 50a | Input | Specify the code of the company creating this transaction here. If you leave this value null, your account's default company will be used instead. |
| date | 10a | Input | The date on the invoice, purchase order, etc. , |
| salesPersnCd | 10a | Input | The client application salesperson reference code. |
| customerCode | 50a | Input | The client application customer reference code. |
| custUseType | 25a | Input | The client application customer or usage type. For a list of available usage types, see https://developer.avalara.com/avatax/handling-tax-exempt-customers/ |
| discount | 15p | Input | The discount amount to apply to the document. This value will be applied only to lines that have the discounted flag set to true. If no lines have discounted set to true, this discount cannot be applied |
| purchOrdNo | 50a | Input | Purchase Order Number for this document This is required for single use exemption certificates to match the order and invoice with the certificate. |
| exemptNo | 25a | Input | Exemption Number for this document. |
| shipFrmline1 | 50a | Input | Ship From Address line 1 |
| shipFrmline2 | 50a | Input | Ship From Address line 2 |

| | | | |
|---|---|---|---|
| shipFrmline3 | 50a | Input | Ship From Address line 3 |
| shipFrmCity | 50a | Input | Ship From City |
| shipFrmRegn | 2a | Input | Ship From State or province name or abbreviation |
| shipFrmCntry | 3a | Input | Ship From Country Code |
| shipFrmPstCd | 10a | Input | Ship From Postal or ZIP code |
| shipFrmLat | 10p | Input | Ship From Latitude |
| shipFrmLong | 10p | Input | Ship From Longitude |
| shipToline1 | 50a | Input | Ship To Address line 1 |
| shipToline2 | 50a | Input | Ship To Address line 2 |
| shipToline3 | 50a | Input | Ship To Address line 3 |
| shipToCity | 50a | Input | Ship To City |
| shipToRegn | 2a | Input | Ship To State or province name or abbreviation |
| shipToCntry | 3a | Input | Ship To Country Code |
| shipToPstCd | 10a | Input | Ship To Postal or ZIP code |
| shipToLat | 10p | Input | Ship To Latitude |
| shipToLong | 10p | Input | Ship To Longitude |
| parameters | 96a | Input | Special parameters for this transaction. |
| rptLocCode | 50a | Input | Sets the sale location code (Outlet ID) for reporting this document to the tax authority. |
| commit | 5a | Input | Causes the document to be committed if true. This option is only applicable for invoice document types, not orders. |
| batchCode | 10a | Input | BatchCode for batch operations. |
| taxOvrType | 25a | Input | Specifies a tax override for the entire document |
| taxOvrAmt | 15p | Input | Override amount |
| taxOvrDate | 10a | Input | Override date |
| taxOvrRsn | 25a | Input | Override reason |
| currencyCde | 3a | Input | Currency Code - 3 character ISO 4217 |

| serviceMode | 25a | Input | Specifies whether the tax calculation is handled Local, Remote, or Automatic (default). This only applies when using an AvaLocal server. = ['Automatic', 'Local', 'Remote'] string Enum: "Automatic", "Local", "Remote" |
|---|---|---|---|
| exchangeRate | 15p | Input | Currency exchange rate from this transaction to the company base currency. |
| exRteEffDate | 10a | Input | Effective date of the exchange rate |
| posLaneCode | 10a | Input | Sets the POS Lane Code sent by the User for this document. |
| busIdNumber | 25a | Input | AT business identification number for the customer for this transaction. This number will be used for all lines in the transaction, except for those lines where you have defined a different business identification number. If you specify a VAT business identification number for the customer in this transaction and you have also set up a business identification number for your company during company setup, this transaction will be treated as a business-to-business transaction for VAT purposes and it will be calculated according to VAT tax rules |
| isSellrIOR | 5a | Input | Specifies if the Transaction has the seller as IsSellerImporterOfRecord |
| description | 256a | Input | Description |
| email | 50a | Input | e-mail address |
| debugLevel | 15a | Input | If the user wishes to request additional debug information from this transaction, specify a level higher than 'normal' = ['Normal', 'Diagnostic'] |
| id | 20i | Output | The unique ID number of this transaction |
| companyId | 20i | Output | The unique ID number of the company that recorded this transaction |
| paymentDate | 10a | Output | The date when payment was made on this transaction. By default, this should be the same as the date of the transaction |
| taxDate | 10a | Output | The date on which this transaction occurred |
| status | 10a | Output | The status of the transaction = ['Temporary', 'Saved', 'Posted', 'Committed', 'Cancelled', 'Adjusted', 'Queued', 'PendingApproval', 'Any'] |
| custVndCode | 25a | Input | Customer Vendor Code |

| reconciled | 5a | Output | If this transaction has been reconciled against the company's ledger, this value is set to true |
|---|---|---|---|
| locationCode | 10a | Input | If this transaction was made from a specific reporting location, this is the code string of the location. For customers using Returns, this indicates how tax will be reported according to different locations on the tax forms |
| totAmount | 15p | Output | The total amount of this transaction |
| totExempt | 15p | Output | The amount of this transaction that was exempt |
| totTax | 15p | Output | The total tax calculated for all lines in this transaction |
| totTaxable | 15p | Output | The portion of the total amount of this transaction that was taxable |
| totTaxCalcd | 15p | Output | If a tax override was applied to this transaction, indicates the amount of tax Avalara calculated for the transaction |
| adjReason | 35a | Input | If this transaction was adjusted, indicates the unique ID number of the reason why the transaction was adjusted. = ['NotAdjusted', 'SourcingIssue', 'ReconciledWithGeneralLedger', 'ExemptCertApplied', 'PriceAdjusted', 'ProductReturned', 'ProductExchanged', 'BadDebt', 'Other', 'Offline'] |
| adjDesc | 128a | Input | If this transaction was adjusted, indicates a description of the reason why the transaction was adjusted |
| locked | 5a | Output | If this transaction has been reported to a tax authority, this transaction is considered locked and may not be adjusted after reporting. |
| region | 2a | Output | The two character ISO region code of the region for this transaction |
| country | 2a | Output | The two-character ISO 3166 code of the country for this transaction |
| version | 5i | Output | If this transaction was adjusted, this indicates the version number of this transaction. Incremented each time the transaction is adjusted. |
| softwareVer | 15a | Output | The software version used to calculate this transaction |
| originAddrId | 20i | Output | The unique ID number of the origin address for this transaction |
| destAddrId | 20i | Output | The unique ID number of the destination address for this transaction |

| modifiedDate | 24a | Output | The date/time when this record was last modified |
| --- | --- | --- | --- |
| modifiedBy | 20i | Output | The user ID of the user who last modified this record |

| Field Name | Size/Type | Input or Output | Description |
|---|---|---|---|
| **Tax Document Lines** | | | |
| | | | |
| **TaxLin** | | | |
| id | 20i | Output | The unique ID number of this transaction |
| transId | 20i | Output | The unique ID number of the company that recorded this transaction |
| lineNumber | 10a | Input | Line number within this document |
| bndryOvrId | 20i | Output | The unique ID number of the boundary override applied to this line item |
| custUseType | 25a | Input | The client application customer or usage type. For a list of available usage types, see https://developer.avalara.com/avatax/handling-tax-exempt-customers/ |
| Description | 200a | Input | A description of the item or service represented by this line |
| shipFrmline1 | 50a | Input | Ship From Address line 1 |
| shipFrmline2 | 50a | Input | Ship From Address line 2 |
| shipFrmline3 | 50a | Input | Ship From Address line 3 |
| shipFrmCity | 50a | Input | Ship From City |
| shipFrmRegn | 2a | Input | Ship From State or province name or abbreviation |
| shipFrmCntry | 3a | Input | Ship From Country Code |
| shipFrmPstCd | 10a | Input | Ship From Postal or ZIP code |
| shipFrmLat | 10p | Input | Ship From Latitude |
| shipFrmLong | 10p | Input | Ship From Longitude |
| shipToline1 | 50a | Input | Ship To Address line 1 |
| shipToline2 | 50a | Input | Ship To Address line 2 |
| shipToline3 | 50a | Input | Ship To Address line 3 |
| shipToCity | 50a | Input | Ship To City |
| shipToRegn | 2a | Input | Ship To State or province name or abbreviation |
| shipToCntry | 3a | Input | Ship To Country Code |
| shipToPstCd | 10a | Input | Ship To Postal or ZIP code |
| shipToLat | 10p | Input | Ship To Latitude |
| shipToLong | 10p | Input | Ship To Longitude |
| discAmount | 15p | Output | The amount of discount that was applied to this line item. This represents the difference between list price and sale price of the item. In general, a discount represents money that did not change hands; tax is calculated on only the amount of money that changed hands |
| discTypeId | 20i | Input | The type of discount, if any, that was applied to this line item |
| exemptAmt | 15p | Input | The amount of this line item that was exempt |
| exemptCertId | 20i | Input | The unique ID number of the exemption certificate that applied to this line item |
| exemptNo | 25a | Output | If this line item was exempt, this string contains the word 'Exempt' |

| isItmTaxable | 5a | Output | True if this item is taxable |
|---|---|---|---|
| isSSTP | 5a | Output | True if this item is a Streamlined Sales Tax line item |
| itemCode | 50a | Input | Item Code (SKU) |
| lineAmount | 15p | Input | The total amount of the transaction, including both taxable and exempt. This is the total price for all items. To determine the individual item price, divide this by quantity. |
| Quantity | 15p | Input | The quantity of products sold on this line item |
| ref1 | 50a | Input | Reference 1 - Client specific reference field |
| ref2 | 50a | Input | Reference 2 - Client specific reference field |
| reportDate | 10a | Output | The date when this transaction should be reported. By default, all transactions are reported on the date when the actual transaction took place. In some cases, line items may be reported later due to delayed shipments or other business reasons |
| revenueAcct | 50a | Input | The revenue account number for this line item |
| Sourcing | 15a | Output | Indicates whether this line item was taxed according to the origin or destination. = ['Mixed', 'Destination', 'Origin'] |
| Tax | 15p | Output | The amount of tax generated for this line item |
| taxableAmt | 15p | Output | The taxable amount of this line item |
| taxCalcd | 15p | Output | The tax calculated for this line by Avalara. If the transaction was calculated with a tax override, this amount will be different from the "tax" value |
| taxCode | 25a | Input | The code string for the tax code that was used to calculate this line item |
| taxCodeId | 20i | Output | The unique ID number for the tax code that was used to calculate this line item |
| taxDate | 10a | Output | The date that was used for calculating tax amounts for this line item. By default, this date should be the same as the document date. In some cases, for example when a consumer returns a product purchased previously, line items may be calculated using a tax date in the past so that the consumer can receive a refund for the correct tax amount that was charged when the item was originally purchased |
| taxEngine | 35a | Output | The tax engine identifier that was used to calculate this line item |
| exemptCode | 25a | Input | Exemption number for this line |
| discounted | 5a | Input | True if the document discount should be applied to this line |
| taxIncluded | 5a | Input | True if tax was included in the purchase price of the item. |
| busIdNumber | 25a | Input | VAT business identification number used for this transaction |
| taxOvrType | 25a | Input | If a tax override was specified, this indicates the type of tax override. = ['None', 'TaxAmount', 'Exemption', 'TaxDate', 'AccruedTaxAmount', 'DeriveTaxable'] |
| taxOvrAmt | 15p | Input | If a tax override was specified, this indicates the amount of tax that was requested |
| taxOvrDate | 10a | Input | Tax Override Date |
| taxOvrRsn | 25a | Input | Tax Override Reason |
| parameters | 96a | Output | Contains a list of extra parameters that were set when the transaction was created |

| Field Name | Size/Type | Input or Output | Description |
|---|---|---|---|
| **Tax Document Addresses** | | | |
| | | | |
| **TaxAdr** | DS | | |
| id | 20i | Output | The unique ID number of this address |
| transId | 20i | Output | The unique ID number of the document to which this address belongs |
| boundaryLev | 15a | Output | The boundary level at which this address was validated. = ['Address', 'Zip9', 'Zip5'] |
| line1 | 50a | Output | Address line 1 |
| line2 | 50a | Output | Address line 2 |
| line3 | 50a | Output | Address line 3 |
| city | 50a | Output | City |
| region | 2a | Output | State or province name or abbreviation |
| postalCode | 10a | Output | Country Code |
| country | 3a | Output | Postal or ZIP code |
| latitude | 10p | Output | Latitude |
| longitude | 10p | Output | Longitude |

| Field Name | Size/Type | Input or Output | Description |
|---|---|---|---|
| **Tax Document Detail** | | | |
| | | | |
| **TaxDet** | | | |
| id | 20i | Output | The unique ID number of this tax detail |
| transId | 20i | Output | The unique ID number of the line within this transaction |
| country | 2a | Output | The two character ISO 3166 country code of the country where this tax detail is assigned |
| region | 2a | Output | The two-or-three character ISO region code for the region where this tax detail is assigned |
| exemptAmt | 15p | Output | The amount of this line that was considered exempt in this tax detail |
| jurisCode | 35a | Output | The code of the jurisdiction to which this tax detail applies |
| jurisName | 50a | Output | The name of the jurisdiction to which this tax detail applies |
| stateAssNo | 35a | Output | The state assigned number of the jurisdiction to which this tax detail applies |
| jurisType | 35a | Output | The type of the jurisdiction to which this tax detail applies. = ['STA', 'CTY', 'CIT', 'STJ', 'CNT'] |
| nonTaxAmt | 15p | Output | The amount of this line item that was considered nontaxable in this tax detail |
| rate | 9p | Output | The rate at which this tax detail was calculated |
| tax | 15p | Output | The amount of tax for this tax detail |
| taxableAmt | 15p | Output | The taxable amount of this tax detail |
| taxType | 35a | Output | The type of tax that was calculated. Depends on the company's nexus settings as well as the jurisdiction's tax laws. = ['Lodging', 'Bottle', 'ConsumerUse', 'Excise', 'Fee', 'Input', 'Nonrecoverable', 'Output', 'Rental', 'Sales', 'Use'] |
| taxName | 50a | Output | The name of the tax against which this tax amount was calculated |
| taxAuthTypId | 20i | Output | The type of the tax authority to which this tax will be remitted |
| taxCalcd | 15p | Output | The amount of tax that was calculated. This amount may be different if a tax override was used. If the customer specified a tax override, this calculated tax value represents the amount of tax that would have been charged if Avalara had calculated the tax for the rule |
| rateType | 35a | Output | (DEPRECATED) The rate type for this tax detail. Please use rateTypeCode instead. = ['ReducedA', 'ReducedB', 'Food', 'General', 'IncreasedStandard', 'LinenRental', 'Medical', 'Parking', 'SuperReduced', 'ReducedR', 'Standard', 'Zero'] |

| Field Name | Size/Type | Input or Output | Description |
|---|---|---|---|
| **Tax Summary** | | | |
| | | | |
| **TaxSum** | | | |
| country | 2a | Output | Two character ISO-3166 country code |
| region | 2a | Output | Two or three character ISO region, state or province code, if applicable |
| jurisType | 35a | Output | The type of jurisdiction that collects this tax. = ['Country', 'Composite', 'State', 'County', 'City', 'Special'] |
| jurisCode | 35a | Output | Jurisdiction Code for the taxing jurisdiction |
| jurisName | 50a | Output | The name of the jurisdiction that collects this tax |
| taxAuthType | 35a | Output | The unique ID of the Tax Authority Type that collects this tax |
| stateAssNo | 35a | Output | The state assigned number of the jurisdiction that collects this tax |
| taxType | 35a | Output | The tax type of this tax. = ['Lodging', 'Bottle', 'ConsumerUse', 'Excise', 'Fee', 'Input', 'Nonrecoverable', 'Output', 'Rental', 'Sales', 'Use'] |
| taxName | 50a | Output | The name of the tax |
| taxGroup | 50a | Output | Group code when special grouping is enabled |
| rateType | 35a | Output | (DEPRECATED) Indicates the tax rate type. Please use rateTypeCode instead. = ['ReducedA', 'ReducedB', 'Food', 'General', 'IncreasedStandard', 'LinenRental', 'Medical', 'Parking', 'SuperReduced', 'ReducedR', 'Standard', 'Zero'] |
| taxable | 15p | Output | Tax Base - The adjusted taxable amount. |
| rate | 9p | Output | Tax Rate - The rate of taxation, as a fraction of the amount. |
| tax | 15p | Output | Tax amount - The calculated tax (Base * Rate). |
| taxCalcd | 15p | Output | Tax Calculated by Avalara AvaTax. This may be overriden by a TaxOverride.TaxAmount. |
| nonTaxable | 15p | Output | The amount of the transaction that was non-taxable. |
| exempt | 15p | Output | The amount of the transaction that was exempt |

| Field Name | Size/Type | Input or Output | Description |
|---|---|---|---|
| **Tax Error Messages** | | | |
| | | | |
| **TaxMsg** | | | |
| code | 32a | Output | Message identifier |
| message | 64a | Output | A summary of what this message tells us |
| description | 128a | Output | Detailed information that explains what the summary provided |
| refersTo | 64a | Output | Information about what object in your request this message refers to |
| severity | 32a | Output | A category that indicates how severely this message affects the results |

# Appendix E: Managing Digital Certificates

This process will guide you through setting up the Digital Certificate Manager to enable your IBM i to interact as a client to other external servers requiring SSL connections.

1. Note that 5722AC3 (Crypto Access Provider 128-bit) or equivalent must be installed as a LICPGM first.

2. Make sure your HTTP *ADMIN server is running. You can verify that the *ADMIN instance is started by issuing the following command:

```
WRKSBSJOB SBS(QHTTPSVR)
```

3. You can start the *ADMIN server by issuing this command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

4. Open Windows Explorer and type http://<iseriesipaddress>:2001 in the address bar. This should bring up the navigator sign on screen. Sign on as a security officer.

   For release V7R1 and greater if the above does not work then try this format.

   http://<Your IBM IP Address>:2001/QIBM/ICSS/Cert/Admin/qycucm1.ndm/main0
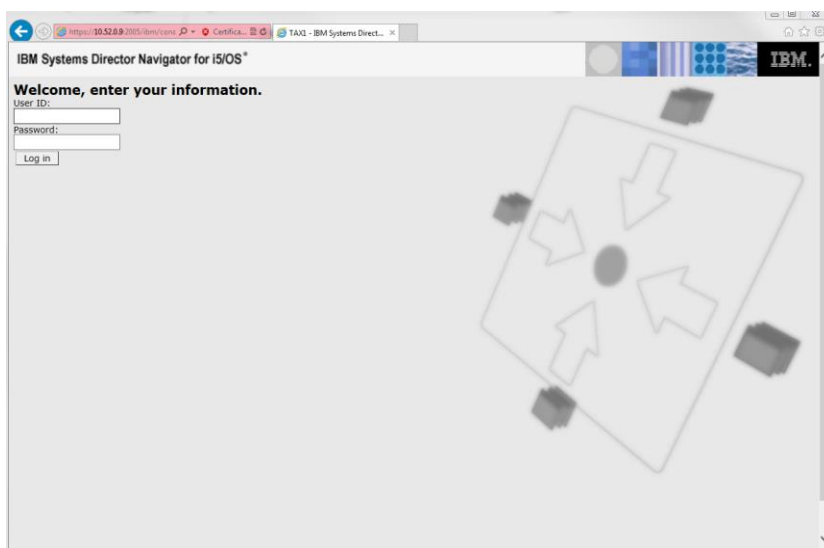
## Select internet configurations
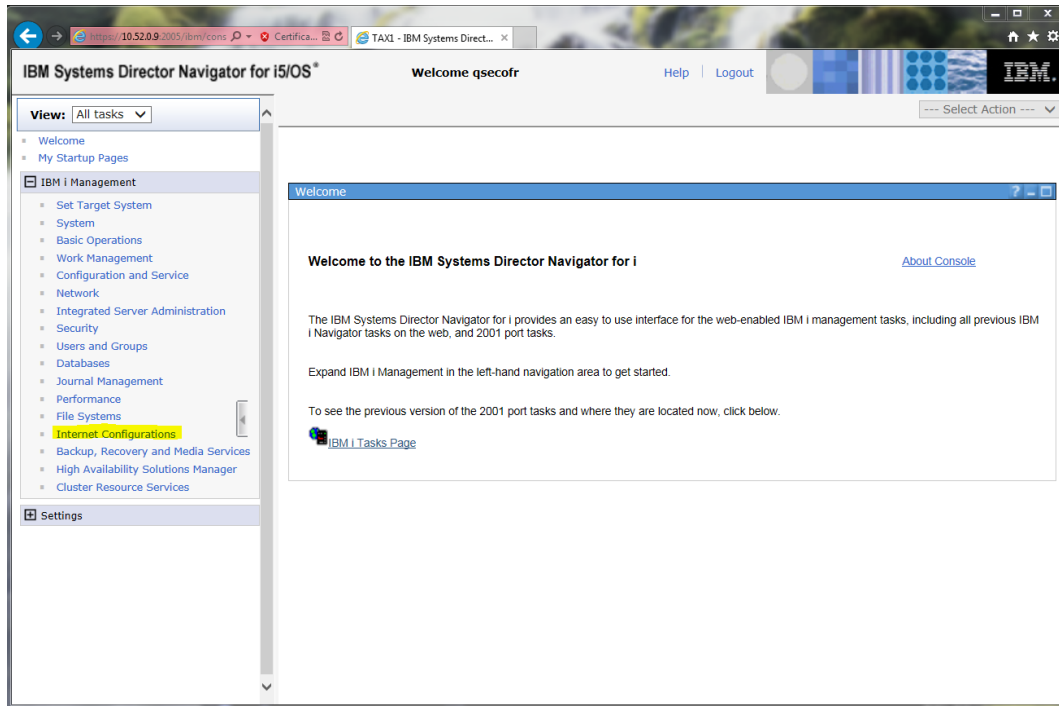


## Select Digital Certificate Manager
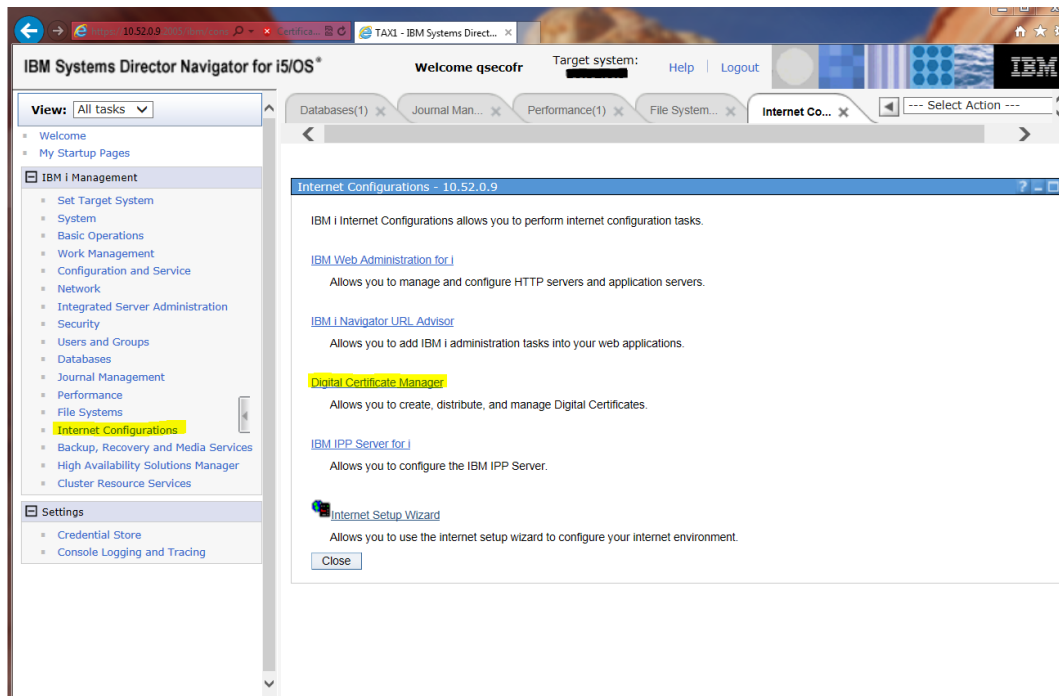
# Try to create a New Certificate store



If you already have a *System  store then you will not see *SYSTEM in the selection list. If you do see it (like the example below) create a new one. Select *SYSTEM and hit continue.
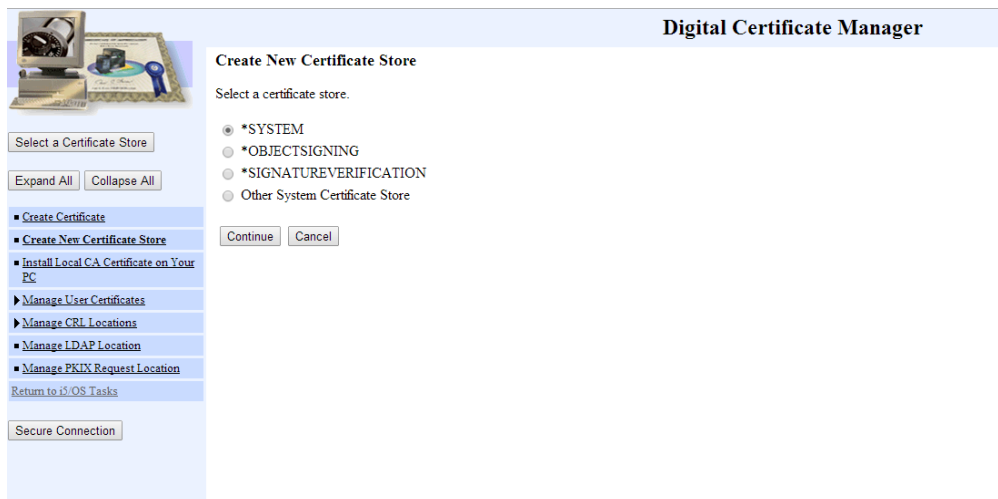


Click 'No – Do not create a certificate in the certificate store' and continue. The system will create the store with a selection of certificates from the major Cert Authorities, which should be sufficient to connect with the Avalara service. Then follow the prompts to create certificate store.

Once your *System store is created you can view it using the Work with Certificates option. The certificate authority you need is the Verisign Class 3 shown below.
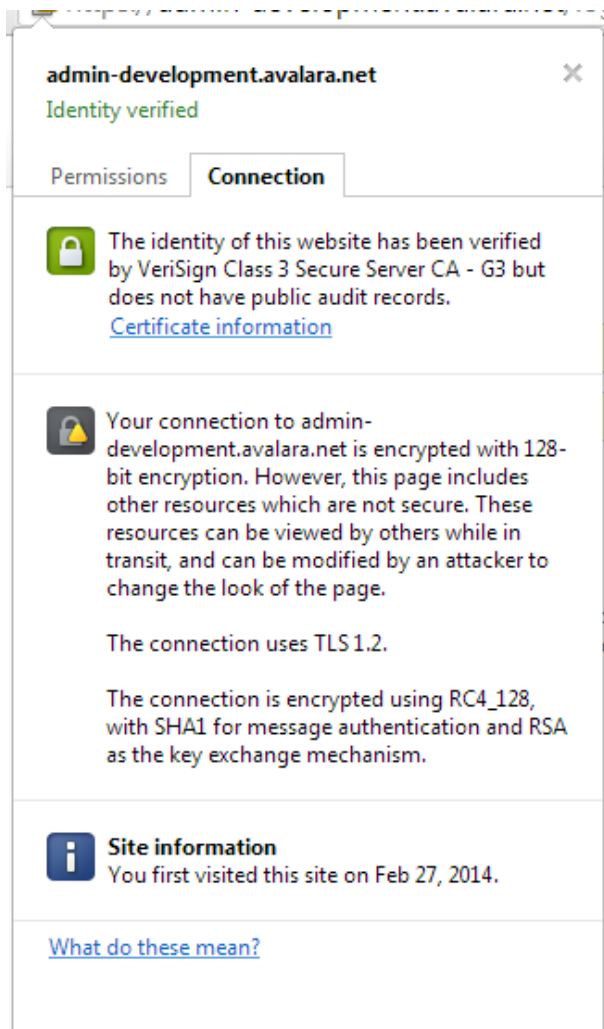
That should be all you need to do for setting up the certificate store apart from ensuring that your users have authorization access.

If, by any chance, you do not see the Verisign Class 3 CA in your store like the example above you will need to import it following the instructions below.

Go to our development console login page. Note the green padlock in the address bar.
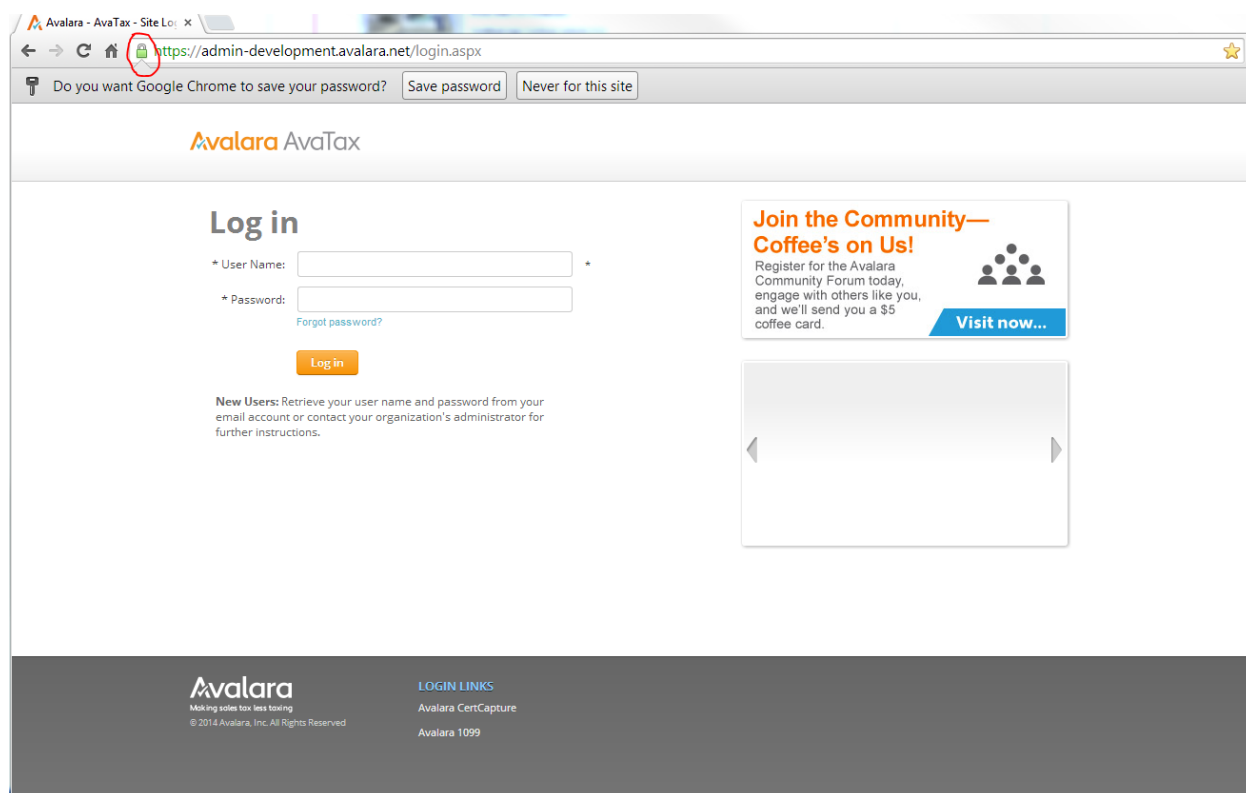
Right click on the padlock and select the connection tab to display the drop down below. Click on Certificate information.



admin-development.avalara.net ✕
Identity verified

Permissions | **Connection**

🔒 The identity of this website has been verified by VeriSign Class 3 Secure Server CA - G3 but does not have public audit records.
Certificate information

⚠ Your connection to admin-development.avalara.net is encrypted with 128-bit encryption. However, this page includes other resources which are not secure. These resources can be viewed by others while in transit, and can be modified by an attacker to change the look of the page.

The connection uses TLS 1.2.

The connection is encrypted using RC4_128, with SHA1 for message authentication and RSA as the key exchange mechanism.

ℹ **Site information**
You first visited this site on Feb 27, 2014.
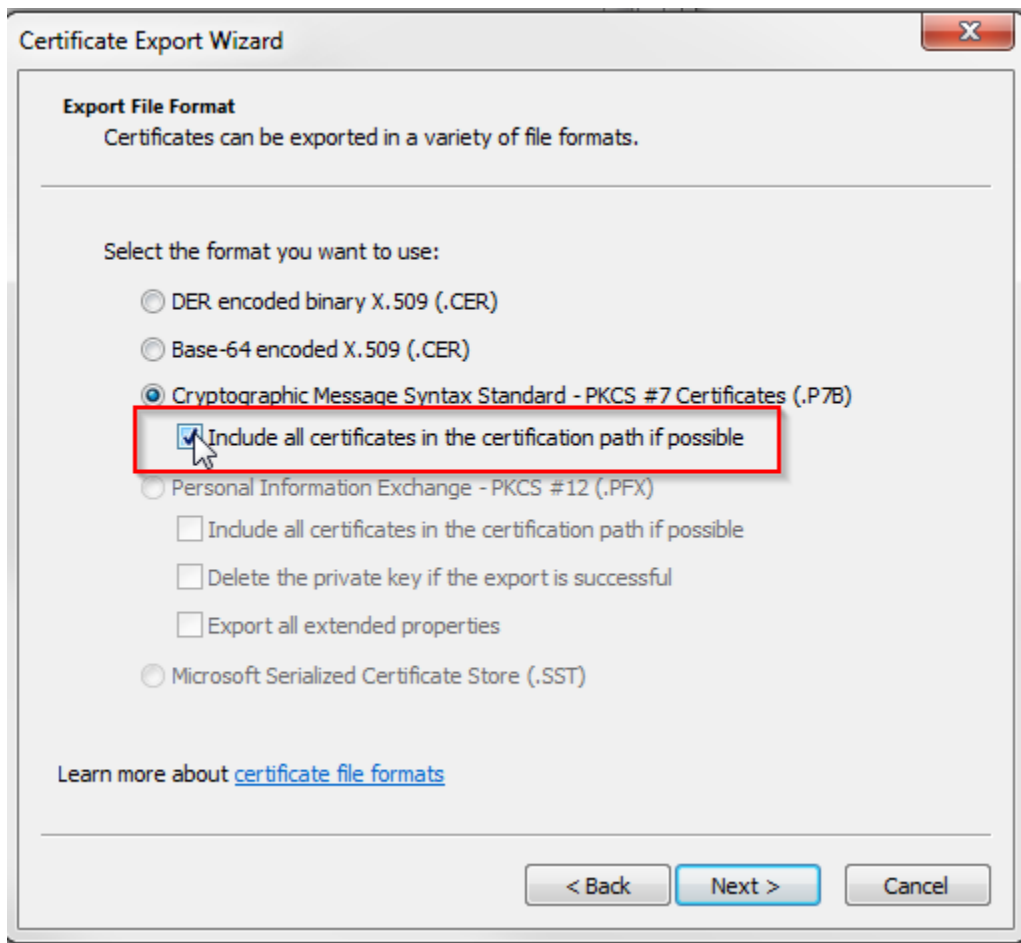
What do these mean?

Click on the Details tab and then the 'Copy to File' button. This will start the Certificate Export Wizard.
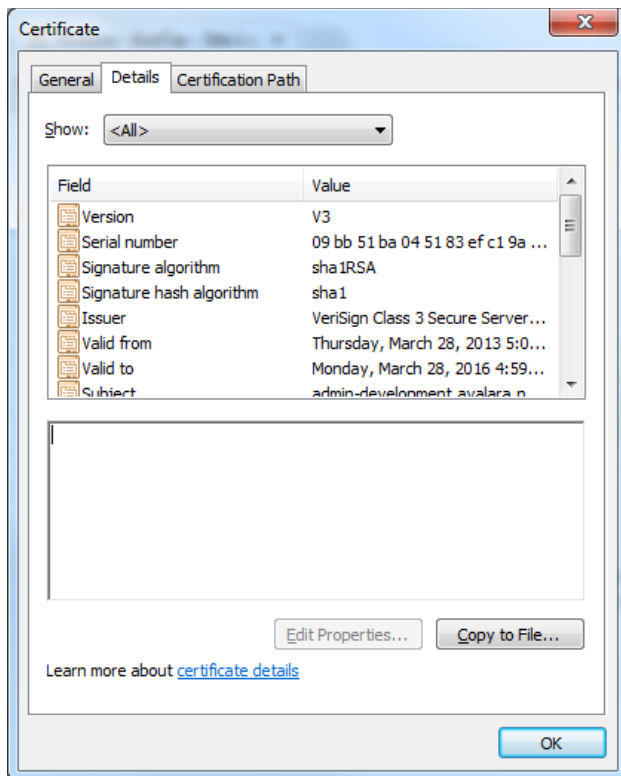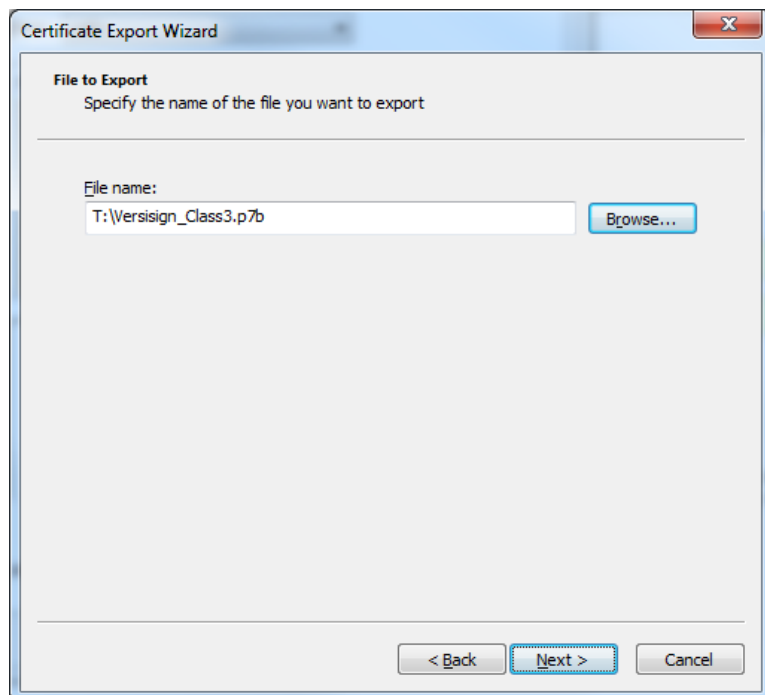
Click Next >

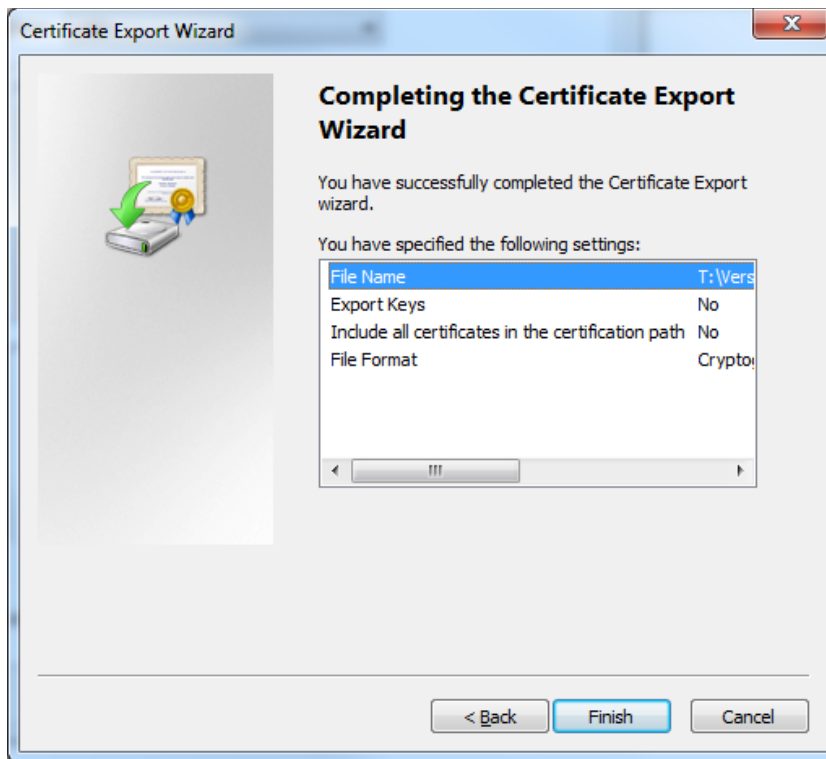Select the .P7B format and Include all. Hit Next >



Give the file a location and save name (we recommend saving in the /tmp directory of your IFS).
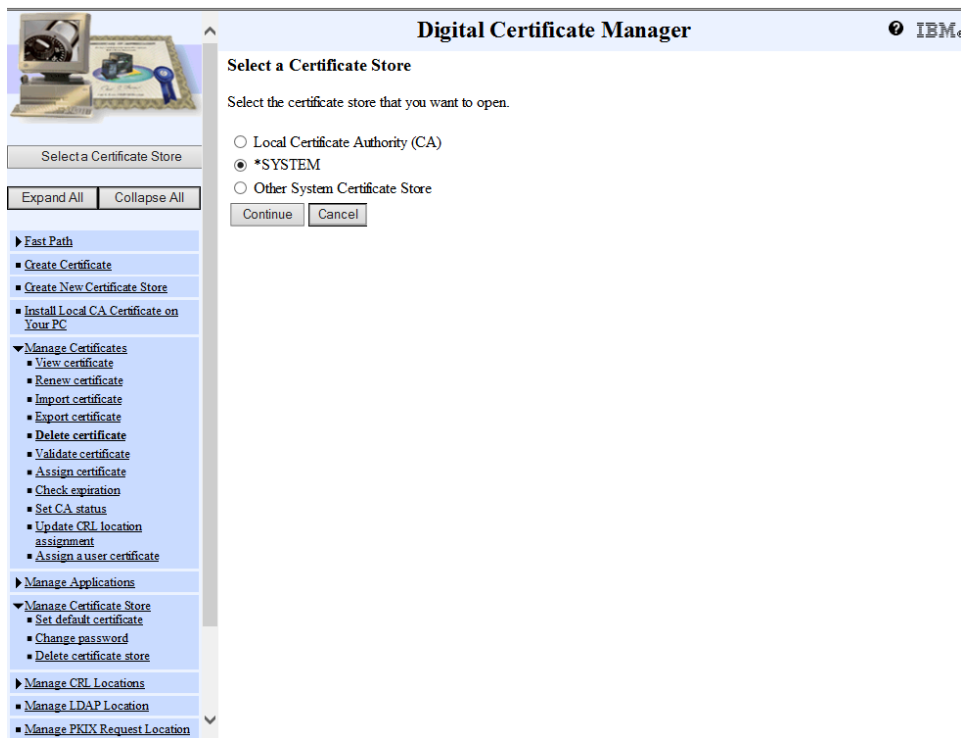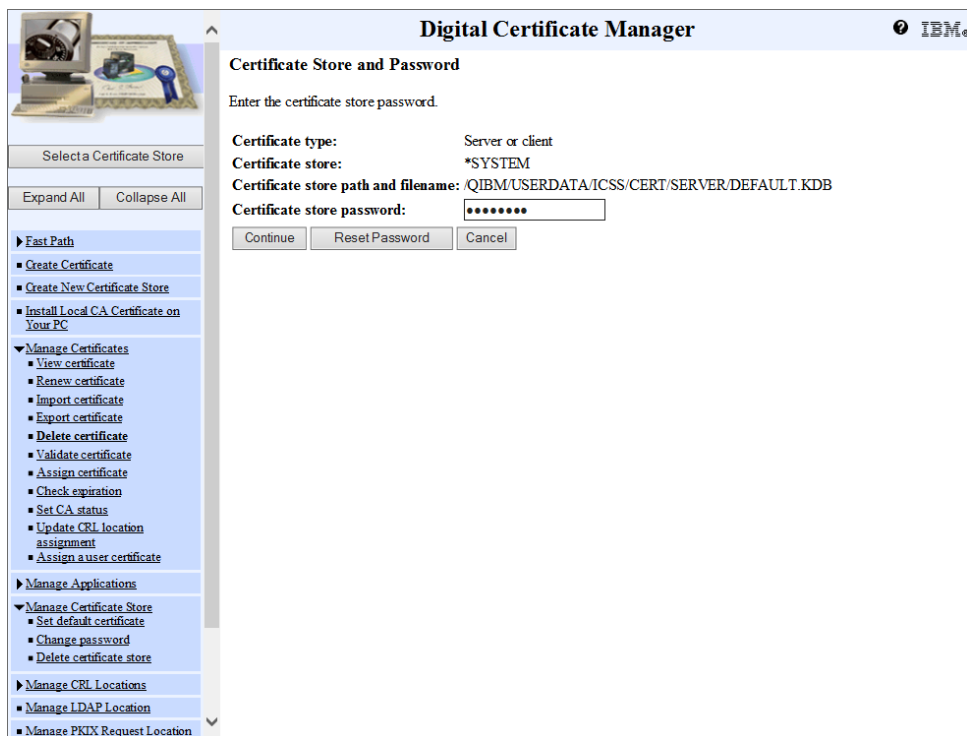
Hit next >

Hit Finish and you should get an 'Export Successful' message.
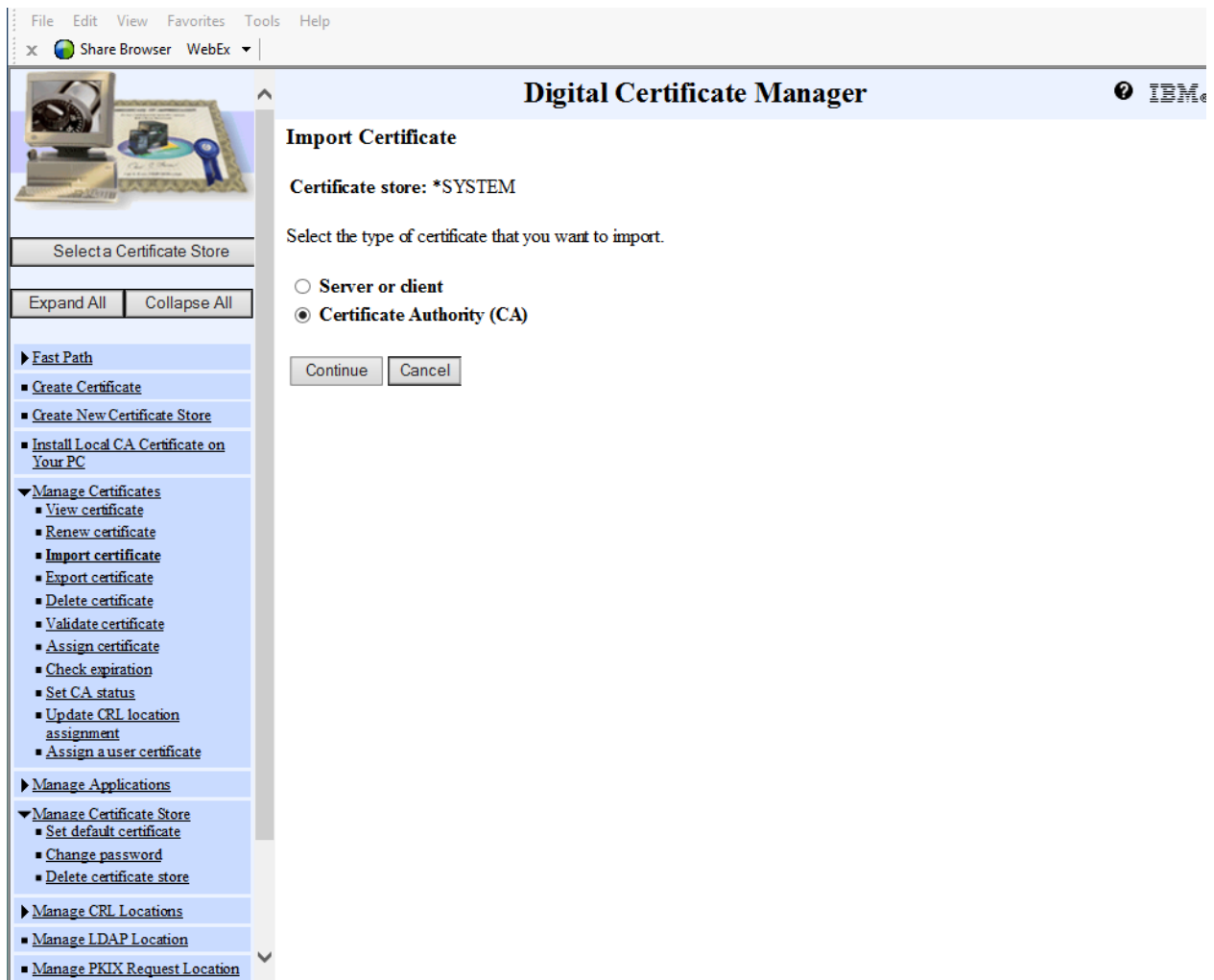
Now go back to DCM and select the *SYSTEM store



Enter your password


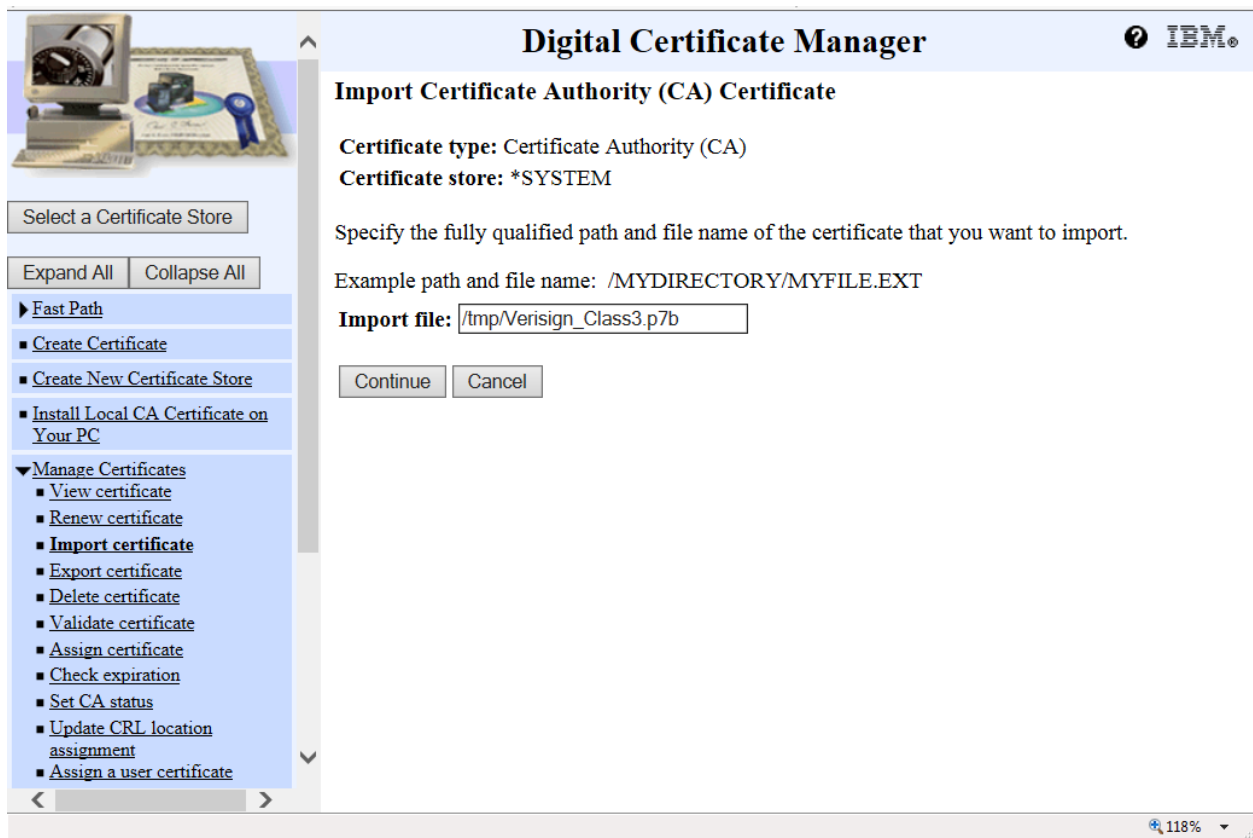
Select Import Certificate > Certificate Authority (CA)

File    Edit    View    Favorites    Tools    Help

✕    🌐 Share Browser    WebEx    ▾

# Digital Certificate Manager    ❓ IBM.

Select a Certificate Store

| Expand All | Collapse All |

▸ Fast Path
- Create Certificate
- Create New Certificate Store
- Install Local CA Certificate on Your PC
▾ Manage Certificates
  - View certificate
  - Renew certificate
  - **Import certificate**
  - Export certificate
  - Delete certificate
  - Validate certificate
  - Assign certificate
  - Check expiration
  - Set CA status
  - Update CRL location assignment
  - Assign a user certificate
▸ Manage Applications
▾ Manage Certificate Store
  - Set default certificate
  - Change password
  - Delete certificate store
▸ Manage CRL Locations
- Manage LDAP Location
- Manage PKIX Request Location

## Import Certificate

**Certificate store: *SYSTEM**

Select the type of certificate that you want to import.

○ **Server or client**
◉ **Certificate Authority (CA)**

| Continue |    | Cancel |

---

Enter the location and name of the import file that you saved above.



**Digital Certificate Manager**    IBM®

**Import Certificate Authority (CA) Certificate**

**Certificate type:** Certificate Authority (CA)
**Certificate store:** *SYSTEM

Specify the fully qualified path and file name of the certificate that you want to import.

Example path and file name:  /MYDIRECTORY/MYFILE.EXT

**Import file:** /tmp/Verisign_Class3.p7b

Continue    Cancel

Navigation sidebar:

Select a Certificate Store

Expand All    Collapse All
▶ Fast Path
■ Create Certificate
■ Create New Certificate Store
■ Install Local CA Certificate on Your PC
▼ Manage Certificates
  ■ View certificate
  ■ Renew certificate
  ■ **Import certificate**
  ■ Export certificate
  ■ Delete certificate
  ■ Validate certificate
  ■ Assign certificate
  ■ Check expiration
  ■ Set CA status
  ■ Update CRL location assignment
  ■ Assign a user certificate

118%

Specify a label … you may want to use **Verisign Class 3 Public Primary Certification Authority** which matches what we have. Hit Continue.
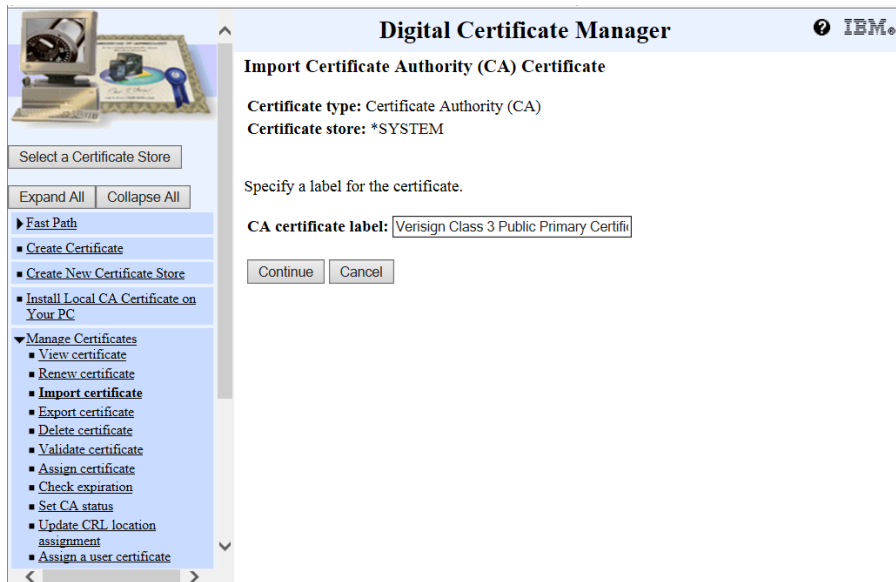


If it all works ok you should get a confirmation message like this.
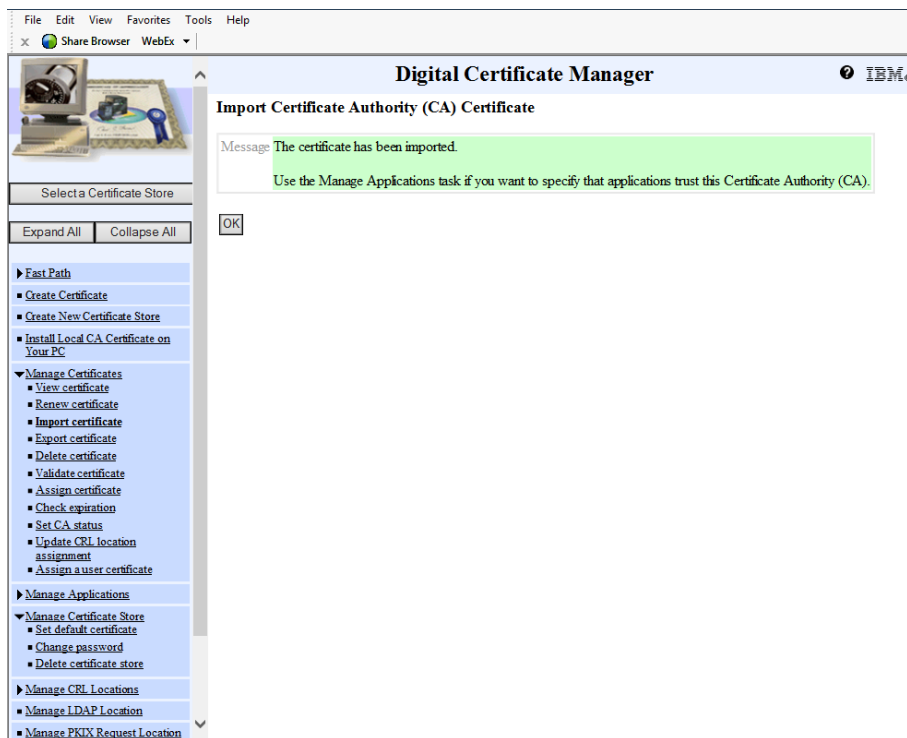
# Appendix F: Address Validation Service

Avalara also provides an address validation service for USA and Canada addresses. An address validation SDK is included with your install library.

The address service can also be called separately, perhaps at the time that you enter or modify an address within your system. See the source for RPGLE program RV2006R as an example. This program can be seen running from the Inquiry menu.

RV2010R is a simple skeleton application that can be used to quickly develop a program that will read through your address database and cleanse the addresses. This will correct errors in names or zips, add zip+4, convert case to a standard format etc. To use this program look at the source in AVATAXR2/QRPGLESRC RV2010R and follow the instructions in the program.

# Appendix G: Entity Control File

Maintenance option 2. Maintain Entity Control file gives you access to a table who's primary function is to provide a conversion between IS0 3166 alpha-3 codes and alpha-2 codes, The SDK address and tax adapters automatically convert an incoming country code to ISO 3166-2 at runtime. This is the format used by Avalara.

By default, the table is set up with only alpha-2 codes. If your calling application passes alpha-3 codes you can configure the table, via the maintenance option, to create a conversion between alpha-3 and alpha-2. Below is an example of how you would configure the table to handle the USA ISO 3166-3 code. The US entry was copied and the ISO3 value set to USA. Now whenever USA is passed as the country code the adapter will convert it to US before calling the service.

```
AC004R1                 Maintain Entity Control File              11/04/14
                                                                  15:12:33

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


     US    _    <= Position                              Cur  Addr
 Opt ISO3 Typ ISO2 Entity Name                           Cde  Chk?
  _   US    1   US    United States                       USD  N
  _   USA   1   US    United States                       USD  N
  _   UY    1   UY    Uruguay                             UYU  N
  _   UZ    1   UZ    Uzbekistan                          UZS  N
  _   VA    1   VA    Vatican City                        EUR  N
  _   VC    1   VC    Saint Vincent and the Grenadines    XCD  N
  _   VE    1   VE    Venezuela                           VEF  N
  _   VG    1   VG    British Virgin Islands              USD  N
  _   VI    1   VI    U.S. Virgin Islands                 USD  N
  _   VN    1   VN    Vietnam                             VND  N
  _   VU    1   VU    Vanuatu                             VUV  N
  _   WF    1   WF    Wallis and Futuna                   XPF  N
  _   WS    1   WS    Samoa                               WST  N
  _   XXX   1   US    United States                       USD  N
                                                             More...

 F3=Exit    F6=Add
 Copyright 2004-2014 Avalara, Inc. All Rights Reserved
```

The Entity Control file also contains currency code information.

If you are using address validation at tax calculation time (controlled by the flag on the System Control file) then you can fine tune the level of validation performed by setting the Address Check flag in this file. So, for example, you could perform address validation for the United States but not for Canada, or you could choose to only validate addresses in California.

| Version | Date | Appendix H: Change Log |
|---------|------|------------------------|
| V1.0 | July 2017 | Initial Release |
| V1.1 | July 2017 | Changed the format of the TaxSum, TaxAdr, TaxMsg and TaxLin data structures. Formerly the were addressed using this format TaxLin(i).taxrate. This has now changed to TaxLin.taxrate(i).<br><br>This resolves an issue with debug being unable to handle the previous form of the TaxLin data structure. The other structures were changed for consistency. |
| V1.2 | July 2017 | Increased field size of the meta data "id" fields. Some customers were experiencing "receiver too small" errors. |
| V1.3 | July 2017 | Enhancements to installer program. |
| V1.4 | Sept 2017 | Minor changes to the Integration Tester |
| V1.5 | Sept 2017 | Corrected errors in documentation relating to field size specification. Removed ExemptNo from the TaxLin data structure – use exemptCode instead. Increased the size of Taxhdr. modifiedDate to 24A to accommodate full date/timestamp. |
| V1.6 | Oct 2017 | Fixed issue where adapter would not calculate tax if it was not in debug mode.<br><br>Added 3 'Easy Operations'. These pre-built functions make it simple to retrieve the jurisdiction breakdown for a Canadian transaction, to Void a document and to Commit a document. See the section on 'Easy Operations' for more details. |
| V1.7 | Nov 2017 | Added a feature to enhance time out resilience. The tax adapter now traps time out conditions, sets the time out value to 30 seconds and retries sending the document 5 times. If there is no success after 5 attempts a time out error is returned to the caller. See the section on '**Time Outs**' for more details.<br><br>Added an additional Easy Operation to retrieve a previously submitted documents details. See the section **Retrieve Document Details** for more information. |
| V1.8 | Dec 2017 | This includes enhancements to the time out feature. Addresses the condition where the results of a 'create' transaction are not returned to the caller due to a time out. However, the transaction was successfully created in Avatax and subsequent attempts to send the transaction again result in a '**DocStatus is invalid for this operation.**' Error. The adapter will now automatically detect that a time out has occurred and that details were not returned to the caller. In this situation, it will attempt to retrieve the transaction details and return them to the caller who will see it as a normal transaction. If after 5 attempts the adapter cannot retrieve the details it will return a time out error.<br><br>This release also includes a feature that allows a developer to force a time out situation which will allow them to test their error handling capabilities. See the section on '**Time Outs**' for more details. |
| V1.9 | Jan 2018 | Modified the SDK adapter to use the 'createoradjust' version of the create operation. For more details go to …<br><br>https://developer.avalara.com/api-reference/avatax/rest/v2/methods/Transactions/CreateOrAdjustTransaction/ |
| V1.10 | Feb 2018 | 1) This version of the SDK should play happily with any other versions of the HTTPAPI open source software that you might have on your system.<br><br>2) Includes a new 'Easy Operation' for creating a full invoice refund. See the section in this document covering Easy Operations. |

| V1.11 | Apr 2018 | Fixed bug where TaxSum.exempt() value was not being passed back to the caller. |
|---|---|---|
| V1.12 | Apr 2018 | Added more error feedback from the JSON parser. In the event of an unexpected parser error the SDK adapter will return the following information to the caller.<br><br>// Error Return Code set to 'Error'<br>  sys.errorCode = 'Error'<br>// Severity set to 'Fatal'<br>  sys.errorSev = 'Fatal';<br>// Program that detected error<br>  sys.errorPgm = 'RV2001R';<br>// Return an error message that includes which part of the JSON response that failed to parse plus the document Id that it failed on<br>  sys.errorText = 'Unexpected error parsing the header details for DocId: 0123456' |
| V1.13 | May 2018 | Fixed minor bug with the System Control file where if the service had the active flag set to 'N' and the user performed a 'T' test then the program crashed with the following error - Error: "RNQ0121 Array index is out of range program RV2001R" . |
| V1.14 | May 2018 | Fixed bug which was preventing line level addresses from overriding those specified at the header level. |
| V1.15 | July 2018 | Customer reported the following sporadic error.<br><br>Message . . . . . : The target for a numeric operation is too small to hold the result (C G D F).<br>Cause . . . . . : RPG procedure RV2001R in program AVATAXR2/RV2001R at statement 038200 performed an arithmetic operation which resulted in a value that is too large to fit in the target. If this is a numeric expression, the overflow could be the result of the calculation of some intermediate result.<br><br>This was happening in the routine that prints stats to the debug log. This release fixes that issue. |
| V1.16 | Aug 2018 | Sends the X-Avalara-Client: header specification. F8 option from System Control Maintenance allows editing of the client string.  See link below for details:<br><br>https://developer.avalara.com/avatax/client-headers/?referrer=&lastReferrer=developer.avalara.com&sessionId=1534538075521 |
| V1.17 | Nov 2018 | Fixed an issue with the System Control file maintenance program where Company Codes greater that 25 chars in length were not being handled properly.<br><br>This resulted in a '**Update or delete in file RV2001P without prior input**' error.<br><br>This did not affect  customers using a company code less than 25 characters. |
| V1.18 | Dec 2018 | Maintenance: Updated the SDK to use the latest version of the YAJL JSON parser. |
| V1.19 | Dec 2018 | Enhancement: The SDK can now be installed into multiple libraries not just AVATAXR2. For example, this allows users to have AVATAXR2 as their production library running, say release V1.5, and then download the latest release, say V1.9, into a test library like AVATAXTEST for testing before upgrading production. |

| | | The BUILDR2 command now accepts a library name to install into. If that library does not exist then it is created as a fresh install, if it does exist then it is upgraded and all custom settings are retained. |
|---|---|---|
| V1.20 | Jan 2019 | A customer reported having some issues when using the SDK on a system that contains the open source library LIBHTTP. This release isolates the SDK from the programs in LIBHTTP and ensures that it runs its own custom versions.<br><br>IMPORTANT NOTE: If you have been following the example program RV2EXAMPLE then you will need to change the H-specs for any connector program that have created so that it uses the new binding directory names (shown below) before compiling it.<br><br>```<br>************** Beginning of data **************<br>0001.00 // ****************************************************<br>0002.00 // * STEP 1 - include the following H-Specs<br>0003.00 // ****************************************************<br>0004.00 //<br>0005.00 H DFTACTGRP(*NO) BNDDIR('AHTTPAPI')<br>0006.00 H BNDDIR('AYAJL') DECEDIT('0.')<br>0007.00 H OPTION(*SrcStmt:*NoDebugIO)<br>``` |
| V1.21 | Feb 2019 | 1) The Request Filename in the System Control file has been modified to accept the *Auto feature in the same way as it can be used for the Response Filename. However, Avalara recommends that customers leave this field blank. This forces the SDK to build requests in main memory rather than in an external file, which is both faster and more reliable. However, if you do have a circumstance where you want to preserve the outgoing JSON request for other uses then you should use *Auto. See the section entitled Configuring **the System Control File** for more details.<br><br>2) The TaxSum data structure has been re-dimensioned to Dim(600) as some customers were exceeding the original value. |
| V1.22 | Mar 2019 | A new flag in the System Control file (Return Basic Tax Details Only) now gives the customer the ability to have the SDK adapter return basic tax details only. This improves performance and reliability for those customers who only require access to the following header data.<br><br>taxDate     The date on which this transaction occurred<br>totAmount    The total amount of this transaction<br>totExempt    The amount of this transaction that was exempt<br>totTax      The total tax calculated for all lines in this transaction<br>totTaxable   The portion of the total amount of this transaction that was taxable<br><br>          If a tax override was applied to this transaction, indicates the amount of tax<br>totTaxCalcd  Avalara calculated for the transaction<br><br>IMPORTANT NOTE: If you have been following the example program RV2EXAMPLE then you will need to add the following line of code to step 5, after the chain to RV2001P before recompiling your connector.<br><br>```<br>SysCtl.basicOnly = ctl_bsic;<br>``` |

| V1.23 | Mar 2019 | 1) The SDK adapter was upgraded to use the version 1.39 of the HTTPAPI open source software (previously using 1.32). One customer was reporting the following error with HTTPAPI 1.32. |
|---|---|---|

RNX0100  Escape               50  03/20/19  11:19:13.483320  QRNXIE     QSYS     *STMT   AVLRAPIR4  AVATAXR2   *STMT

    From user . . . . . . . . . :   QTMHHTP1

    From module . . . . . . . . :   QRNXMSG

    From procedure  . . . . . . :   SignalException

    Statement . . . . . . . . . :   25

    To module . . . . . . . . . :   COMMTCPR4

    To procedure  . . . . . . . :   COMMTCP_FD_SET

    Statement . . . . . . . . . :   4951

    Message . . . . :   Length or start position is out of range for the string

     operation.

    Cause . . . . . :   One of the following has occurred in RPG procedure

     COMMTCP_FD in program AVATAXR2/AVLRAPIR4: - A numeric length or start

     position is less than 1 or too large for the string operation. - The

     search-argument parameter of the %SCAN built-in function has zero length or

     is longer than the source-string parameter. - The maximum-length parameter

     of the %STR built-in function is not a value between 1 and the maximum size

     of a character field. Recovery  . . . :   Contact the person responsible for

     program maintenance to determine the cause of the problem.

See https://archive.midrange.com/web400/201901/msg00083.html for more details.

    2) The Address service adapter (which uses the 'resolve' operation) was modified to accept the *Auto feature for the response file and the debug log.

```
RV2001R              Edit System Control  File Record              3/21/19
                                                                  12:52:58

   Company: 1_____        Active: Y


   Operation: resolve
              (create, adjust, commit, void , resolve, transactions)
   Authorization String (BASE64 encoded):
   _____._____


      Account Type: S (S-andbox or P-roduction)

   Request Filename: _____ (Opt)
    Result Filename: /tmp/*Auto )_____  (Opt)

   Compact JSON Y/N: Y           Avatax Request Time Out (secs): ___10

                              Return Basic Tax Details Only: N

   Debug Y/N: Y Debug Filename: /tmp/*Auto_____
                                (e.g /tmp/mydebug.txt)



   F3=Exit F12=Cancel
```

| | | |
|---|---|---|
| V1.24 | Jul 2019 | A customer using the SDK with a web server application was having intermittent issues which were traced to the open source HTTPAPI software used by the SDK.<br><br>The following article describes a similar issue that was resolved by resizing one of the HTTPAPI variables.<br><br>http://www.scottklement.com/archives/ftpapi/201504/msg00096.html ,<br><br>The original person who reported the issue changed the size of field "`fdset' in PRIVATE_H from 28a to 64a" in order to correct it.  It was noted that "The error only occurs in CGI Web server jobs."<br><br>In this version of the SDK fdset is re-sized to 200a |
| V1.25 | Jul 2019 | Increased the decimal places for TaxDet.rate  and TaxSum.rate to be 6 places instead of 4 |
| V1.26 | Aug 2019 | Modified the way that the SDK uses activation groups. The SDK tax and address adapters have been changed to use *CALLER which means that it will run in the group used by the calling program. This means that you can specify you own activation group in your connector, if you wish. By default the SDK demo programs and functions use the activation group name AVATAXR2 for tax calls and AVATAXR2A for address validation calls. |
| V1.27 | Sep 2019 | Made improvements to the Address Validation Adapter. Removed some unnecessary code that may have been causing some customers occasional issues. Improved the error handling routine to pass back more information about a failed validation call. Increased the size of the error return field SvcErr.Details(n) from 128A to 256A to fully capture messages returned from the Avatax address API. |

| V1.28 | Sep 2019 | Fixed an issue with the Address Validation adapter ('resolve' operation) where lat/long co-ordinates were not being returned to the caller. |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------|
| V1.29 | Oct 2019 | At the request of a customer the debug file numeric suffix has been increased from a max value of 999 to 99999. |
| V1.30 | Feb 2020 | Enhanced the Refund - Easy Operation to allow it to perform the following types of refund.<br><br>• A full refund of a previous sale<br>• Refunding just the tax that was charged on a previous sale. (useful when the customer provides an exemption certificate after the purchase)<br>• Refunding one or more items (lines) from a previous sale<br>• Granting a customer a percentage refund of a previous sale<br><br>See page 35 for more details. |

## 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.