



# AvaTax SDK for IBM i

---

## Implementation Guide

Version 14.1.15

Revision date: 11/30/2018

*Avalara may have patents, patent applications, trademarks, copyrights, or other intellectual property rights governing the subject matter in this document. Except as expressly provided in any written license agreement from Avalara, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

© 2015 Avalara, Inc. All rights reserved.

*Avalara, AvaTax Calc, AvaTax Returns, AvaTax Certs, AvaRates, TrustFile, BPObridge, Laserbridge+, Sales TaxII, Sales TaxPC, SalestaxPC+, StatetaxII, and StatetaxPC are either registered trademarks or trademarks of Avalara, Inc. in the United States or other countries.*

*All other trademarks are property of their respective owners.*

## Contents

Introduction.....	4
The Business Case .....	4
Document Scope .....	4
Who Should Read this Document .....	5
Features and Benefits of the AvaTax SDK for IBM i .....	5
What Does the AvaTax SDK for IBM i Do? .....	5
Prerequisites .....	6
Requirements and Constraints .....	6
Software and Hardware Requirements.....	6
Digital Certificates .....	7
General Constraints .....	11
XML Data Restrictions .....	11
AvaTax Admin Console .....	13
AvaTax Admin Console and Web Service.....	14
Web Service Access .....	15
Client Profile Information .....	16
Installing the AvaTax SDK for IBM i .....	18
Install the AVATAX Library onto your IBM Server.....	18
Authorizing Users to the AvaTax SDK for IBM i Menu System .....	24
Configure the System Control File .....	27
GetTax Example .....	28
Testing the Connection to Avalara Web Services .....	32
To Test the Connection to Avalara Web Services .....	32
SDK Integration Tester .....	35
Integration Tester Defaults .....	36
Canadian Taxes .....	38
Developing Your Connector.....	40
Web Service Operations.....	40
Ping .....	41
GetTax .....	41
PostTax / CommitTax .....	43
CancelTax .....	43
AdjustTax .....	43
Interfacing with AvaTax SDK for IBM i.....	43

HTI - HTTP Parameters Used by the Web Service Call.....	44
AXI – Order/Invoice Parameters .....	44
AXO – Tax Data Returned by AvaTax.....	44
JUO – Tax Jurisdiction Data Returned by AvaTax .....	44
Required Fields .....	45
Common System Messages .....	46
Coding Considerations.....	48
Other Considerations .....	49
Tax Compliance .....	60
Nexus Jurisdictions .....	61
ItemCode versus TaxCode .....	62
Pre - GoLive Check List .....	63
Documents .....	63
Item Codes and Description .....	63
Nexus Jurisdictions.....	63
Server Audit.....	63
How to Go Live .....	64
Appendix A: HTTP Parameters.....	66
Appendix B: Order/Invoice Parameters .....	67
Appendix C: Tax Data Parameters .....	72
Appendix D: Tax Jurisdiction Data Parameters .....	77
Appendix E: Managing Digital Certificates.....	79
Appendix F: Address Validation Service.....	93
Appendix G: Entity Control File .....	94
Appendix H: Change Log .....	95
Appendix J – Upgrading from release 13 to 14.....	98
Avalara Support .....	103

## 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 situs<sup>1</sup>, nexus, tax tiers, tax holidays, exemptions, certificate management, and product taxability rules.

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

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

## The Business Case

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

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

## Document Scope

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

---

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

## Who Should Read this Document

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

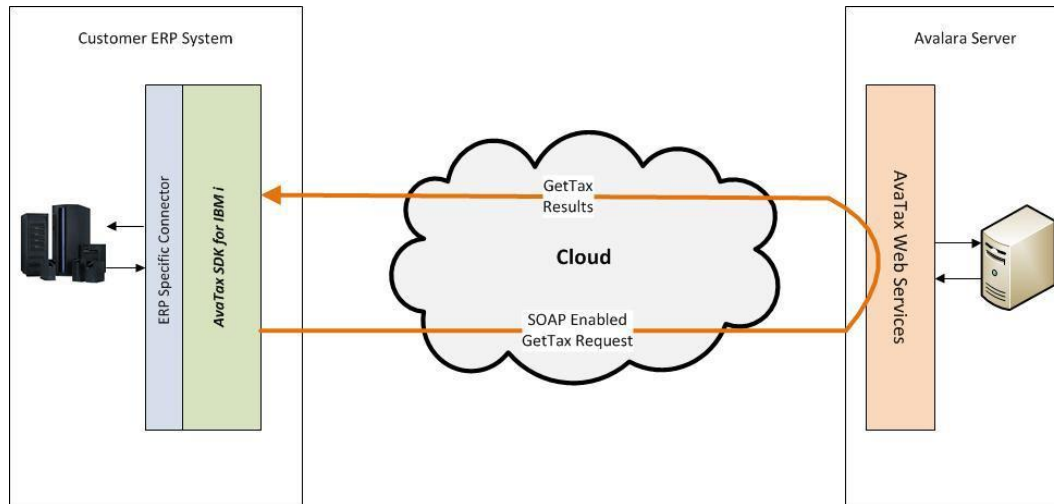


Figure 1.1

## Features and Benefits of the AvaTax SDK for IBM i

The SDK adapter:

- Is written entirely in RPG. There is no need for developers to use or understand Java. Furthermore, there is no need to implement IBM Websphere.
- Uses W3C-recommended Simple Object Access Protocol (SOAP) 1.1 and WSDL best practices.
- Runs on all versions of the System i operating system from V5R4.
- Has a small footprint. It requires the installation of a single library of about 150MB.
- Handles all Internet communications with Avalara web services.
- Is supported by Avalara.

## What Does the AvaTax SDK for IBM i Do?

A typical use would be as follows:

1. An application passes an invoice to the SDK adapter for which a tax calculation is required.
2. The SDK adapter receives the invoice details and generates a SOAP XML request which it transmits over the Internet to the Avalara 'GetTax' web service.
3. The SDK adapter then receives the response message from the service, parses it, and returns the calculated tax results back to the calling application. 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 adapter. Check that these are satisfied before beginning installation.

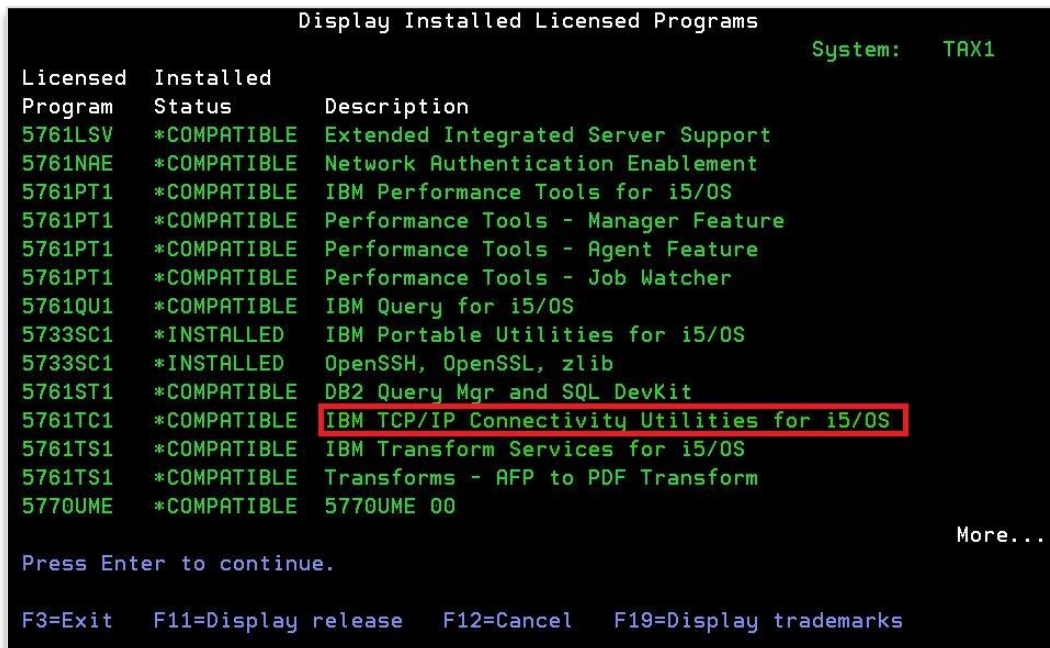
### Software and Hardware Requirements

- A System i (AS/400, iSeries, or i5) running OS/400 version V5R4 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

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

```

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

--Object Authorities--
Opt  User      Data 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

Opt  User          Data          --Object Authorities--
      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

Opt  User          Data          --Object Authorities--
      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

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

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

```

Fig.2.1b

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)
```

## General Constraints

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

### Version 1 – for Pre-V6R1M0 installations

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

- 99 detail lines
- 12 addresses

#### **NOTE**

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

*As of September 2016 a "cut-down" version is also available that can handle 350 invoice lines. It does not expose all the API parameters but concentrates on the most commonly used data.*

### Version 2 – for V6R1M0 or newer

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

- 4500 detail lines
- 450 addresses

## XML Data Restrictions

**Note:** From version 14.1.4 the SDK adapter will automatically convert XML illegal characters and some foreign character sets (e.g. French characters) and so no action is required on the part of the caller. However, if you do run into an issue with a character that cannot be converted then please read the section below.

The IBM i SDK uses XML to format messages for the Avatax service. You should be aware that the ampersand character (&) and the left angle bracket (<) are illegal characters for XML and MUST NOT appear in their literal form, except when used as markup delimiters.

What sometimes happens is that a client will pass a string to the service that includes a '&' character resulting in an error. For example, the address below with the & on line 1 .....

```

AX005R1                      Address Validation Tester                      4/25/14
                                                                           09:21:47

      Enter address to check                      Validated Result
Line1: Smith & Jones llc
Line2: 100 Ravine Lane
Line3: _____
City: Bainbridge Island
State: WA
Zip: 98110

Case: Default (Default, Upper, Mixed)

                                Other Details
                                County:
                                FIPS:
                                Carrier Route:
                                Post Net:
                                Address Type:
                                Latitude:
                                Longitude:

F3=Exit F4=Clear
Copyright 2004-2014 Avalara, Inc. All Rights Reserved
MA  a                               MW                               18/053

```

.... results in the error

```

AX005R1                      Address Validation Tester                      4/25/14
                                                                           09:21:47

.....
:                               Tax Adapter Message                       :
:                                                                           :
:   Result: Error                                                         :
:   Summary:                                                         :
:   Details: Unexpected error - possible invalid character(s) in address line :
:                                                                           :
:                                                                           :
:   HelpLnk:                                                         :
:   RefersTo:                                                         :
:   Severity:                                                         :
:   Source:                                                           :
:                                                                           :
:   F12=Cancel                                                         :
:                                                                           :
:.....

F3=Exit F4=Clear
Copyright 2004-2014 Avalara, Inc. All Rights Reserved
MA  a                               MW                               04/004

```

If you have the type of address database that may include these characters then use the RPG Scan and Replace function to convert these characters into something valid before passing to Avatax.

It's a one line conversion in RPG IV from V7R1 onwards ...

```
%SCANRPL(scan string : replacement : source { : scan start { : scan length } )
```

An example of replacing the '&' character would be this .....

```
address_line = "Smith & Jones Ilc"
```

```
address_line = %ScanRpl('&' : 'and' : address_line);
```

Result would be ... address\_line = "Smith and Jones Ilc" ... which would be valid to pass to Avatax.

Pre- V7R1 operating systems can do something similar with %XLATE, %SCAN and %REPLACE.

## 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. 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](mailto: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.

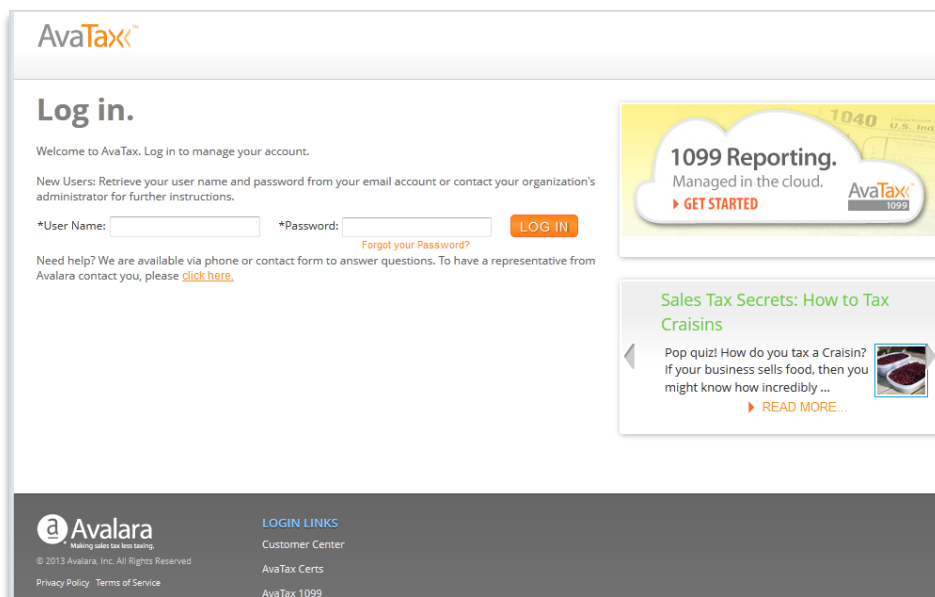


Figure 2.1

## Web Service Access

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 (AX0001P). 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 have entered your production company and credentials and set the Account Type = 'P'.

```

AC001R1                      Edit System Control File Record                      3/08/16
                                                                           09:42:33

Company:  1                      Id: 001                      Service Active: Y

Account:  Graham                      Password: .....

Account Type: S  (S-andbox, P-roduction, C-CloudConnect)

Service Provider: Avalara Inc.          Client: Avatax SDK Development

Adapter: Avatax IBMi SDK 14.1.4      Client System: IBM i

Address Validation: N

CloudConnect URI:
_____

Debug: Y Debug File Name: /tmp/gsw/*Auto      (e.g /tmp/debug.txt)

F3=Exit  F5=Show/Hide Password  F12=Cancel
  
```

MA B 04/012

Figure 2.2

- Company

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

- Account

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

- Password

This needs to set in the password field of your system control file.

- Account Type

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

**https://development.avalara.net** is the URL of the **development account service**. This must be passed in your connector program but in order to make things simpler the correct development URL is pre-populated in your system control file dependent on this flag.

Select 'P' to use a production account.

**https://avatax.avalara.net** is the URL of the **production account service**. When you are ready to go live, you can re-direct to this URL by simply changing value of this field to 'Y' (don't forget to also make sure you are using the correct company and credentials)

Select 'C' to use a CloudConnect device. **Note:** CloudConnect is an Avalara service which provides the user with a local tax calculation device when extreme resilience and/or exceptional performance is required.

If you select this option you will also need to enter the CloudConnect URI

- Address validation

Set to 'Y' if you wish to automatically validate an address before a tax service call. For most users, it is probably a better practice to validate addresses at the point of data entry rather than before each GetTax call. GetTax also performs a certain amount of address validation on its own.

- Address Conversion Type

This is only used if Address Validation is 'Y'. Set to Default, Upper or Mixed.

## Client Profile Information

The Client Profile is a required element in your AvaTax SDK integrations for several good reasons. These values are preset by Avalara.

The values passed in the client profile fields:

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

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

- Service Provider

Set to 'Avalara Inc.'.

- Client

This is set to IBM I SDK plus the current version number. However, depending on the ERP you are connecting to we may change this value (if we are doing the install) or provide you with instructions how to update.



- Adapter version

Maintained by Avalara and shows the current version of the SDK adapter.

- Client System

This is set to IBM i

- Debug

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

- Debug File Name

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

**Note:** Each time the service is called the previous debug log will be overwritten. If you wish to keep logs then use \*Auto as the file name. This will cause the adapter to write the debug log to an unique file in the /tmp directory called avatax\_debugnnn.txt where nnn is a unique system generated number. This will number will cycle when it reaches 999. You can also specify your own directory, if you prefer. For example, a debug filename of /tmp/gsw/\*Auto would create a unique file in the /tmp/gsw/ directory.

## Installing the AvaTax SDK for IBM i

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

**Important Note:** If you are upgrading from version 13 to 14 then special steps are required. Please read Appendix J for details.

### Install the AVATAX Library onto your IBM Server

Components for SDK adapter are contained in a single library, by default called AVATAX. The first step is to install the AVATAX 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 System i and install it.

If this is the first time that you have done this then a new SDK library will be created. We recommend sticking with the system default library name – AVATAX. However, this can be changed, if you wish.

If you want to upgrade your existing version of the SDK the installer works the same way as for a new install. However, all your custom settings are retained in the upgraded version.

#### **NOTE**

- *There are two versions of the AVATAX library available. It is very important that you download the correct one for your IBM operating system. Regardless of which one you use they will both, by default, be restored to your server as AVATAX.*

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

1.0) Sign off the System i

2.0) Sign on as QSECOFR or a profile with \*ALLOBJ authority

3.0) If you are already on version 13.2.4 or higher then add your SDK library (normally AVATAX) to your library list and continue at step 5.0

**ADDLIB LIB(AVATAX)**

(Tip: To check version by inspecting the any of your active System Control file records. Maintenance Menu > Option 1)

AC001R1 Edit System Control File Record 3/08/16 09:42:33

Company: 1 Id: 001 Service Active: Y

Account: Graham Password: .....

Account Type: S (S-andbox, P-roduction, C-CloudConnect)

Service Provider: Avalara Inc. Client: Avatax SDK Development

Adapter: Avatax IBMi SDK 14.1.4 Client System: IBM i

Address Validation: N

CloudConnect URI:

Debug: Y Debug File Name: /tmp/gsw/\*Auto (e.g /tmp/debug.txt)

F3=Exit F5=Show/Hide Password F12=Cancel

MA B 04/012

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

4.1) CRTSAVF FILE(QTEMP/AVASTALLER)

4.2) Start an FTP session by typing **ftp 'ftp.taxrates.com'** at the 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 ftp500\_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**

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

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

- 5) Prompt the installer by typing AVABUILD and pressing F4. If your operating system is pre-V6R1 then answer \*YES to 'Is this a pre-V6R1 install?' otherwise leave the default of \*NO. Enter the name of the library you would like to install or upgrade, the default is AVATAX. Hit <Enter>.

```
Type choices, press Enter.

Is this a pre-V6R1 install? . .  *NO          *NO, *YES
Library to Install or Upgrade .  AVATAX        Character value

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

Bottom

MA  a  MW  05/037
```

- 6) 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 System i). The second part submits a job called AVABUILD which updates your SDK library.

Once AVABUILD has finished running you can confirm the result by consulting the job log. At a command line type ....

### WRKUSRJOB

Place an 8 beside job AVABUILD and hit <Enter> to display spooled files.

```

                                Work with User Jobs                                TAX1
                                05/21/14 12:24:03
Type options, press Enter.
 2=Change  3=Hold  4=End   5=Work with  6=Release  7=Display message
 8=Work with spooled files 13=Disconnect

Opt  Job      User      Type  -----Status-----  Function
 8_  AVABUILD  GWILSON  BATCH  OUTQ
__   GSWA1     GWILSON  INTER  OUTQ
__   GSWA1     GWILSON  INTER  OUTQ
__   GSWA1     GWILSON  INTER  ACTIVE              CMD-WUJ

                                                                Bottom
Parameters or command
===>
F3=Exit    F4=Prompt  F5=Refresh  F9=Retrieve  F11=Display schedule data
F12=Cancel F17=Top     F18=Bottom  F21=Select assistance level

MA  a                               MW                                09/003

```

Place a 5 beside file QPJOBLOG and hit <Enter> to display it.

```

                                Work with Job Spooled Files
Job:  AVABUILD      User:  GWILSON      Number:  213787
Type options, press Enter.
 1=Send  2=Change  3=Hold  4=Delete  5=Display  6=Release  7=Messages
 8=Attributes  9=Work with printing status

Opt  File      Device or  User Data  Status  Total  Current  Copies
 5_  QPJOBLOG  QPRINT   AVABUILD  RDY     41     1       1

                                                                Bottom
Parameters for options 1, 2, 3 or command
===>
F3=Exit    F10=View 3  F11=View 2  F12=Cancel  F22=Printers  F24=More keys

MA  a                               MW                                22/007

```

Place a B in the control field and press <Enter> to go to the bottom of the log.

```

Display Spooled File
File ..... QPJOBLOG                               Page/Line 41/35
Control ..... B                                     Columns 1 - 130
Find .....
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3
CPC2191 Completion      00 05/21/14 11:10:50.470822 QLTDLFIL QSYS      006F QLICLIB QSYS      0503
Message ..... : Object ASIAOBJ in QTEMP type *FILE deleted.
CPF1164 Completion      00 05/21/14 11:10:50.470106 QWTNCE0J QSYS      0140 *EXT      *N
Message ..... : Job 213787/6WILSON/AVABUILD ended on 05/21/14 at 11:10:50;
                  .506 seconds used; end code 0 .
Cause ..... : Job 213787/6WILSON/AVABUILD completed on 05/21/14 at
                  11:10:50 after it used .506 seconds processing unit time. The job had
                  ending code 0. The job ended after 1 routing steps with a secondary ending
                  code of 0. The job ending codes and their meanings are as follows: 0 - The
                  job completed normally. 10 - The job completed normally during controlled
                  ending or controlled subsystem ending. 20 - The job exceeded end severity
                  (ENDSEV job attribute). 30 - The job ended abnormally. 40 - The job ended
                  before becoming active. 50 - The job ended while the job was active. 60 -
                  The subsystem ended abnormally while the job was active. 70 - The system
                  ended abnormally while the job was active. 80 - The job ended (ENDJOBABN
                  command). 90 - The job was forced to end after the time limit ended
                  (ENDJOBABN command). Recovery ..... : For more information, see the Work
                  management topic collection in the Systems management category in the i5/OS
                  Information Center, http://www.ibm.com/systems/i/infocenter/.
                                                                                               Bottom
F3=Exit F12=Cancel F19=Left F20=Right F24=More keys
a MW 17/019

```

Page up one page and if all went well you will see a message similar to the one below telling you that the update was successful.

```

Display Spooled File
File ..... QPJOBLOG                               Page/Line 41/16
Control ..... B                                     Columns 1 - 130
Find .....
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3
because they did not meet the selection requirements. Deleted records were
not copied because the default COMPRESS(*YES) was specified. If this is the
Copy From Import File (CPYFRMIMPF) command, deleted records were copied, and
the number of records not copied were the result of copy errors. Technical
description ..... : If the from-file or to-file member or label
name is *N, the file is a printer, nonlabeled tape, or a spooled inline
file. If the from-file library name is *N, the file is a spooled inline
file. Records are copied based on the following selection requirements: --
Character values (INCCCHAR parameter). -- Field values (INCRFL parameter).
*NONE Command          05/21/14 11:10:50.460643 QCADRV QSYS      0309 AC1001C ORANGE      016D
Message ..... : 16200 - DLCOBJ OBJ((ASIAWORK *LIB *EXCLAD))
*NONE Command          05/21/14 11:10:50.460721 QCADRV QSYS      0309 AC1001C ORANGE      0176
Message ..... : 16300 - SNOPGMMSG MSG('ASIAWORK SDK Library upgraded
                  successfully')
*NONE Information       05/21/14 11:10:50.460754 AC1001C ORANGE      0176 QCMD QSYS      01C8
Message ..... : ASIAWORK SDK Library upgraded successfully
*NONE Command          05/21/14 11:10:50.460790 QCLRTNE QSYS      0050 AC1001C ORANGE      01E6
Message ..... : RETURN /* RETURN due to end of CL program
                  */
                                                                                               More...
F3=Exit F12=Cancel F19=Left F20=Right F24=More keys
a MW 03/022

```

If you do not see the success message you will see an error message explaining what the issue was. Correct this issue and try again or contact [support@avalara.com](mailto: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.

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

7) 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 adapter menu items.

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

### Files:

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

### Program:

- SC0005R – User Authority Maintenance

### Command:

- ASIA – start the menu system



To start the menu security program:

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

```
MAIN                               AvaTax SDK for IBM i                               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

## Configure the System Control File

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

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

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

```

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

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

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

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

```

Figure 3.3

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

## GetTax Example

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

```
AC001R1                      Edit System Control File Record                      3/15/16
                                                                14:26:02

Company: APITrialCompany      Id: 001      Service Active: Y

Account: Your account #      Password: .....

Account Type: S (S-andbox, P-rodution, C-CloudConnect)

Service Provider: Avalara Inc.      Client: Set dataara AC0002DA=ERP

Adapter: Avatax IBMi SDK 14.1.4      Client System: IBM i

Address Validation: N      Address Conversion Type: Upper

CloudConnect URI:
_____

Debug: N Debug File Name: _____ (e.g /tmp/debug.txt)

F3=Exit  F5=Show/Hide Password  F12=Cancel
```

Figure 3.4

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

- **Company**

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



Figure 3.5

- **Id**

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

- **Service Active**

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

**NOTE**

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

- **Account**

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

- **Password**

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

**NOTE**

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

Here is your AvaTax Account Number and License Key. **Please keep this information in a safe place**, as it is required to connect to the AvaTax Calc web service.

Account Name: AvaWilson SDK  
Account Number: 1100012245  
License Key: F308A0D4B2CF795C WebService URL: <https://development.avalara.net/>

Best regards,

AvaTax Implementation Support  
[golive@avalara.com](mailto:golive@avalara.com)

Figure 3.6

- **Account Type**

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

**<https://development.avalara.net>** is the URL of the **development account service**. This must be passed in your connector program but in order to make things simpler the correct development URL is pre-populated in your system control file dependent on this flag.

Select 'P' to use a production account.

**<https://avatax.avalara.net>** is the URL of the **production account service**. When you are ready to go live, you can re-direct to this URL by simply changing value of this field to 'Y' (don't forget to also make sure you are using the correct company and credentials)

Select 'C' to use a CloudConnect device. **Note:** CloudConnect is an Avalara service which provides the user with a local tax calculation device when extreme resilience and/or exceptional performance is required.

If you select this option you will also need to enter the CloudConnect URI

- **Service Provider**

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

- **Client**

By default, this is set to 'Avatax SDK for IBM i'. You may change it once but once changed it will not be editable again. Set to the version of the ERP that you are using. (e.g. Infor XA 9.1)

- **Adapter**

This is informational only and is maintained by Avalara.

- **Client System**

This is set to IBM i and cannot be changed.

- **Address Validation**

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

- **Debug**

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

**Debug File Name**

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

**Note:** Each time the service is called the previous debug log will be overwritten. If you wish to keep logs then use \*Auto as the file name. This will cause the adapter to write the debug log to an unique file in the /tmp directory called avatax\_debugnnn.txt where nnn is a unique system generated number. This will number will cycle when it reaches 999. You can also specify your own directory, if you prefer. For example, a debug filename of /tmp/gsw/\*Auto would create a unique file in the /tmp/gsw/ 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>.

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

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

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

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

MA  a                               MW                               08/003
```



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

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

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

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

MA  a                               MW                               08/003
```

Figure 3.7

3. If you get the message shown in Figure 3.8 then re-check your **Account** and **Password** details.



Figure 3.8

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

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

## SDK Integration Tester

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

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

```

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

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

Service: 001 DocId:          <- Orig ->  <-- New -->
Co: APITrialCompany DocTyp: SalesInvoice DocCde:
DocDate: 2013-04-16 CustNo: TEST01 Dis Amt: Exemp No:
EntUse: PO#: P0000001 SalPerCd: 123 Locn: Ref: Commit: N

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

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

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

Figure 3.9

## Integration Tester Defaults

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

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

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

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

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

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

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

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

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

CustNo: TEST01 PO#: P0000001 SalPerCd: 123

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

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

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

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

## 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. In order to make summarizing Canadian taxes easier Avalara provides a program called AX009R1 which can be used to return a summary of Canadian taxes for a given invoice. This program is included in your AVATAX library. A demo program showing how to use AX009R1 is included as menu option 4 on the INQUIRY menu. The source is in AVATAX/QRPGLESRC AX009R1T.

**Note:** AX009R1 uses the Avalara GetTaxHistory service. So make sure that you activate that function with your company code and valid credentials in your System Control file.

To use follow these 4 steps:

- 1) Include this /copy statement in the program where you wish to retrieve the Canadian taxes.

```
0003.00 /copy AX009CP // Include this /copy member for the Canadian Tax Breakdown
```

- 2) Clear the TAX data structure each time before calling GetCanTax

```
0009.00 // Clear TAX Buffer
0010.00 clear TAX;
```

- 3) Set the company code, document code and document type that you wish to retrieve. Then call the tax summary function, as shown.

```
0005.00 // Pass the Company, Invoice and Document Type you want to retrieve
0006.00 TAX.Company = myCompany; // e.g. '001'
0007.00 TAX.DocCode = myInvoice; // e.g. '001145425'
0008.00 TAX.DocType = mtDocType; // e.g. 'SalesInvoice'
0009.00 // Call the GetCanTax function
0010.00 GetCanTax(TAX);
```

- 4) Check TAX.ErrCode after the call. If this is \*Blanks then the call was good and your summarized data will be available as shown below. If it is not \*Blanks then check the error message and take some other action.

```
0010.01      // Check the return code. If TAX.ErrCode = *Blanks
0011.00      // then the call was good and you can read your results.
0012.00      if TAX.ErrCode = *Blanks;
0012.01          // The following fields will contain your summarized tax data
0015.00          // TAX.GstJurisName      e.g. CANADA
0016.00          // TAX.GstTaxName       e.g. CANADA GST/TPS
0017.00          // TAX.GstRate          e.g. 5.000000
0018.00          // TAX.GstTax           e.g. 30.0000
0019.00          //
0020.00          // TAX.PstJurisName     e.g. ONTARIO
0021.00          // TAX.PstTaxName       e.g. ONTARIO HST
0022.00          // TAX.PstRate          e.g. 8.000000
0023.00          // TAX.PstTax           e.g. 48.0000
0025.00      else;
0025.01      // If the TAX.ErrCode is not *Blanks the there was a problem
0025.02      // and you will need to specify your own action.
0026.00          // TAX.ErrCode      This will contain 'Error'
0027.00          // TAX.ErrText      This will contain a description of the error
0029.00      endif;
```

## Developing Your Connector

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

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

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

## Web Service Operations

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

```

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

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

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

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

MA  a                               MW                                07/003

```



## Ping

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

**Ping ResultCode is: Success Ping Version is: 11.11.1.1**

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

## GetTax

Service Code 001

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

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

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

Document Type	Description
Any	This will return all types of documents.
SalesOrder	Sales Order, estimate, or quote. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of <b>Temporary</b> .
SalesInvoice	The document is a permanent invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of <b>Saved</b> .
PurchaseOrder	Purchase order, estimate, or quote. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of <b>Temporary</b> .
PurchaseInvoice	The document is a permanent invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of <b>Saved</b> .
ReturnOrder	Sales Return Order. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of <b>Temporary</b> .
ReturnInvoice	Sales Return Invoice. The document is a permanent sales return invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of <b>Saved</b> .
InventoryTransferOrder	Inventory Transfer order, estimate, or quote. This is a temporary document type and is not saved in tax history. GetTaxResult will return with a DocStatus of <b>Temporary</b> .
InventoryTransferInvoice	Inventory Transfer Invoice. Document is a permanent invoice; document and tax calculation results are saved in the tax history. GetTaxResult will return with a DocStatus of <b>Saved</b> .
ReverseChargeOrder	This type of document will be returned by the AvaTax service for PurchaseOrder for European countries as destination address with business identification number without tax override option.
ReverseChargeInvoice	This type of document will be returned by the AvaTax service for PurchaseInvoice for European countries as destination address with business identification number without tax override option.

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

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

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

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

**GetTax is:** Success

**DocCode:** 100811153742

**TotalAmount:** 1000

**TotalTax:** 86

**Line:** 1 Tax: 86 TaxCode: P0000000

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

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

#### **NOTE**

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

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

## PostTax / CommitTax

Service Code 003 (Post) and 004 (Commit)

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

## CancelTax

Service Code 005

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

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

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

## AdjustTax

Service Code 007

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

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

## Interfacing with AvaTax SDK for IBM i

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

The following is an explanation of the four data structures.

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

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

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

## AXI – Order/Invoice Parameters

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

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

### **NOTE**

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

## AXO – Tax Data Returned by AvaTax

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

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

## JUO – Tax Jurisdiction Data Returned by AvaTax

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

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

## Required Fields

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

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

## Common System Messages

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

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

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

## Coding Considerations

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

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

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

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

If found then;

Clear Httpi buffer;

Populate the Httpi data structure;

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

Repeat until no more '*Shipped*' headers;

Clear the SDK adapter buffers;

Address No = 0;

Line No =0;

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

Address No = Address No +1;

Populate the Asial Address details for Ship To;

Address No = Address No +1;

Populate the Asial Address details for Ship From;

Read the first Invoice Line (D);

Repeat until no more lines for this Invoice;

Line No = Line No +1;

Populate Asial line level details;

Read next Invoice Line;

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

If Return Code = Error;

Take some error action;

else;

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

end;

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

else;

Take some error action;

end;



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

Member	Type	Description
AX002R1	RPGLE	Tax Rate Look Up
AX003R1	RPGLE	Tax Rate Look Up - Ship From/To
AX005R1	RPGLE	Address Validation Tester
AX006R1	RPGLE	Integration Tester

## Other Considerations

### GetTax

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

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

#### NOTE

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

### PostTax / CommitTax, CancelTax, AdjustTax

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

### Detail Level Enumeration

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

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

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

### Document

GetTax Result: Success

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

### Line

GetTax Result: Success

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

Line: 1 Amount: 1000 Tax: 83.5

ItemCode: AvalItem-001 TaxCode: P0000000

### Tax

GetTax Result: Success

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 84.34

Line: 1 Amount: 1000 Tax: 83.5

ItemCode: AvalItem-001 TaxCode: P0000000

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

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

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

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

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

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

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

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

**PST** = Provincial Sales Tax (appears in AvaTax State)

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

**VAT** = Value Added Tax (appears in AvaTax State)

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

### Tax

GetTax Result: Success

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

DocCode: 34567.2

TotalAmount: 1010

TotalTax: 151

Line: 1 Amount: 1000 Tax: 151.00

ItemCode: Avaltem-001 TaxCode: P0000000

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

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

### AvaTax Document States

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

**Posted** - The last call was to PostTax.

**Committed** - The last call was to CommitTax.

#### NOTE

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

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

Therefore:

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

#### NOTE

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

## Handling Return Invoices

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

### Committed Document – Refund (Return Invoice)

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

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

The invoice has the following line items:

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

### Assumptions:

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

### Steps:

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

### NOTE

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

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

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

#### NOTE

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

- Set the Amt field to a negative dollar amount.

#### NOTE

- Always leave Qty as a positive.

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

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

#### NOTE

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

#### Summary:

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

#### Handling Shipping and Freight

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

## Handling Discounts

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

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

### NOTE

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

Some Examples:

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

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

<input type="checkbox"/> Expand All	Currency in USD											
No	Item	Tax Code	Use Code	Sourcing	Line Amount	Discount	Non-Taxable	Taxable	Tax	TO	JO	Tax Included
1	AvaProduct 1	P0000000	D		25.00	4.76	0.00	20.24	1.75			
2	AvaProduct 2	P0000000	D		35.00	6.67	0.00	28.33	2.43			
3	AvaProduct 3	P0000000	D		45.00	8.57	0.00	36.43	3.12			
4	FREIGHT	FR020200	D		15.00	0.00	0.00	15.00	1.30			

Figure 4.1

### NOTE

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

## Messages – Results / Error

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

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

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

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

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

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

### NOTES

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



## Address Validation - GetTax Method

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

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

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

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

## Understanding Boundary levels

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

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

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

### **NOTE**

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



## Handling Tax Jurisdiction Breakdowns

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

similar to the example shown in Figure 4.2 below.

```
beginning of data
//free
// Example of how to read the jurisdiction
// details from the returned data. This snippet
// reads all the taxing jurisdictions for invoice
// line 1 (i.e. JUO(1)).
i = 1;
do (JUO(1).TaxName(i) = *Blanks) or (i >= 10);
    Country = JUO(1).Country(i);           // e.g. US
    Region = JUO(1).RegionID(i);           // e.g. CO
    JurName = JUO(1).JurisName(i);         // e.g. COLORADO
    TaxName = JUO(1).TaxName(i);           // e.g. CO STATE TAX
    TaxRate = JUO(1).Rate(i)*100;          // e.g. 2.9
    eval(H) TaxAmt = JUO(1).Tax(i);        // e.g. 7.11
    i = i + 1;
enddo;
/end-free
```

Figure 4.2

## Useful Links

### WSDLs

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

#### Development

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

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

#### Production

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

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

### Avalara Admin Console

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

#### Development

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

#### Production

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

### Web Service URLs

These are the development and production web service URLs.

#### Development

<https://development.avalara.net>

#### Production

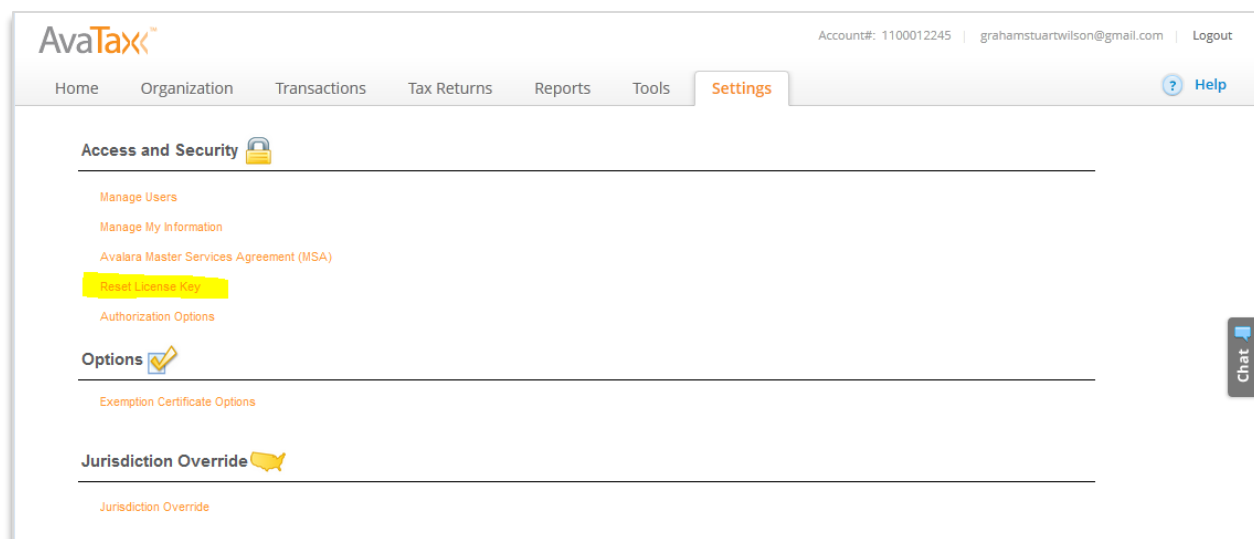
<https://avatax.avalara.net>

### [developer.avalara.com](https://developer.avalara.com)

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

## Resetting Your License Key

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



### **NOTE**

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

## Tax Compliance

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

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

### **NOTE**

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

## Nexus Jurisdictions

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

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

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

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

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

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

### **NOTE**

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

For US-based businesses, AvaTax Calc assumes:

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

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

More granular control over taxing Transactions is provided by

- Tax Codes
- Tax Rules
- Items

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

## ItemCode versus TaxCode

ItemCode versus TaxCode in line Items:

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

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

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

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

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

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

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

## Pre - GoLive Check List

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

### Documents

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

### Item Codes and Description

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

### Nexus Jurisdictions

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

### Server Audit

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

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

## How to Go Live

At some point 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 entries in the System Control file that point to the production rather than the development server.

You will need to change the Account number and Password to be your production credentials supplied to you by Avalara (plus possibly the company code, if different). Also, change 'Is this a PRODUCTION account?' flag to 'Y'.

```
AC001R1                               Edit System Control File Record                               2/12/15
                                                                                               14:30:27

Company:  1                               Id: 001                               Service Active: Y

Account: my_account                       Password: .....

Is this a PRODUCTION account? N

Service Provider: Avalara Inc.             Client: IBM i SDK 13.2.8

Adapter: AVATAX IBMi SDK 13.2.8           Client System: IBM i

Address Validation: N

Debug: N Debug File Name:                  (e.g /tmp/debug.txt)

F3=Exit  F5=Show/Hide Password  F12=Cancel
```

If your test and production systems are on the same System i 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 GetTax service below which has a test company called Test and a production company called 1.



```
AC001R1                Maintain System Control File                1/08/14
                                                                11:20:39
Type option, press Enter.
 2=Edit  3=Copy  4=Delete  5=Display  T=Test

Opt Primary Service      Service Description      Company Code
█ Tax Services           GetTax                Test
- Tax Services           GetTax                1
- Tax Services           PostTax              1
- Tax Services           CommitTax            1
- Tax Services           CancelTax            1
- Tax Services           AdjustTax            1
- Tax Services           Ping                 1
- Address Services       Validate              1
- Address Services       Ping                 1

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

## Appendix A: HTTP Parameters

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

## Appendix B: Order/Invoice Parameters

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

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

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
BINH	25	Char			Optional Business Identification Number	Input	Optional
CancelCode	25	Char			Used by <b>CancelTax</b> . A code indicating the reason the document is getting canceled. Valid values are:  PostFailed DocDeleted DocVoided AdjustmentCancelled	Input	Required
TotalAmt	15	Dec	6		Used by <b>PostTax</b> . The sum of all line Amount values.	Input	Required
TotalTax	15	Dec	6		Used by <b>PostTax</b> . The total tax for the document.	Input	Required
AdjReason	3	Int	0		Used by <b>AdjustTax</b> . Sets a valid reason for the given AdjustTax call. Adjustment Reason is a high level classification of why an Original Document is being modified.  0 Not Adjusted 1 Sourcing Issue 2 Reconciled with General Ledger 3 Exemption Certificate Applied 4 Price or Quantity Adjusted 5 Item Returned 6 Item Exchanged 7 Bad Debt 8 Other (Explain)	Input	Required
AdjDesc	25	Char			Used by <b>AdjustTax</b> . Sets description for the given AdjustTax call. Adjustment Description is required when AdjustmentReason is "Other" for enhanced tractability.	Input	Optional
For Addresses, use the following parameters:							
AddrCode	3	Char		Yes	Programmatically determined value used internally by the adapter. Defaults to the hash code of the Address object. This is Origin or Destination code defined at the document header or line detail level.	Input	Optional
Line1	50	Char		Yes	Address line 1	Input	Optional
Line2	50	Char		Yes	Address line 2	Input	Optional
Line3	50	Char		Yes	Address line 3	Input	Optional
City	50	Char		Yes	City	Input	Optional
Region	2	Char		Yes	State or province name or abbreviation	Input	Optional
PostCode	10	Char		Yes	Postal or ZIP Code	Input	Optional
Country	3	Char		Yes	Country Code (ISO-2 or ISO-3)	Input	Optional
TaxRegId	10	Int	0	Yes	TaxRegionId provides the ability to override the tax region assignment for an address.	Input	Required
Latitude	10	Char		Yes	Latitude	Input	Optional
Longitude	10	Char		Yes	Longitude	Input	Optional
For Line Details, use the following parameters:							

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
No	10	Char		Yes	Line #	Input	Optional
Origin	10	Char		Yes	User assigned code to indicate that an address is to be used as the "Ship From" location for a specific line item. This can be specified at the document or line detail.	Input	Optional
Dest	10	Char		Yes	User assigned code to indicate that an address is to be used as the "Ship To" location for a specific line item. This can be specified at the document or line detail.	Input	Optional
Item	50	Char		Yes	Item Code (SKU)	Input	Optional
TaxCode	25	Char		Yes	System or Custom Tax Code.	Input	Optional
Qty	15	Dec	4	Yes	The quantity represented by this line.	Input	Required
Amt	15	Dec	4	Yes	The total amount for this line item (Qty * UnitPrice).	Input	Required
Disc	5	Char		Yes	Is this line item discounted? 'true' or 'false'	Input	Required
RevAcct	50	Char		Yes	Revenue Account	Input	Optional
Ref1	50	Char		Yes	Client specific reference field	Input	Optional
Ref2	50	Char		Yes	Client specific reference field	Input	Optional
Exempt	25	Char		Yes	Exemption Number for this document	Input	Optional
CustUse	25	Char		Yes	The client application customer or usage type.  The standard values for the CustomerUsageType (A-L) are: A - Federal Government B - State/Local Govt. C - Tribal Government D - Foreign Diplomat E - Charitable Organization F - Religious/Education G - Resale H - Agricultural Production I - Industrial Prod/Mfg. J - Direct Pay Permit K - Direct Mail L - Other	Input	Optional
Desc	200	Char		Yes	Defines the description for the product or item	Input	Optional
OrideTyp	25	Char		Yes	Type of TaxOverride.  None - Default  TaxAmount - The TaxAmount overrides the total tax for the document.  Exemption - Exemption certificates are overridden making the document taxable.  TaxDate - The TaxDate overrides the DocDate as the effective date used for tax calculation.  AccruedTaxAmount - The accrued tax amount.	Input	Optional
OrideAmt	15	Dec	4	Yes	Tax override amount	Input	Optional
OrideDat	10	Char		Yes	The TaxDate overrides the document date (DocDate) as the effective date used for tax calculation.	Input	Optional
OrideRsn	25	Char		Yes	Reason for applying TaxOverride	Input	Optional

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage	Required or Optional
TaxIncl	5	Char		Yes	TaxIncluded indicates the tax amount has already been included in the line amount. The tax will be reversed out and returned in the tax property. The taxable property will indicate the calculated tax base amount.	Input	Optional
BINL	25	Char		Yes	Optional Business Identification Number	Input	Optional

## Appendix C: Tax Data Parameters

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
TransID	20	Char			A unique Transaction ID identifying a specific request/response set.	Output
ResultCode	10	Char			Indicates the success (or SeverityLevel) of the operation.  <b>Success</b> - Operation succeeded. <b>Warning</b> - Warnings occurred, operation succeeded. <b>Error</b> - Errors occurred, operation failed. <b>Exception</b> - Unexpected exceptions occurred, operation failed.	Output
DocID	25	Char			Unique ID returned by GetTax. Can be used as an input to PostTax, CommitTax, CancelTax, or AdjustTax	Output
DocType	25	Char			The original document's type, such as Sales Invoice or Purchase Invoice	Output
DocCode	20	Char			Document Code, that is, the internal reference code used by the client application.	Output
DocDate	10	Char			The date on the invoice, purchase order, etc.	Output
DocStatus	10	Char			The document's status is returned in the GetTaxResult (except for DocStatus.Any) and indicates the state of the document in tax history.  <b>Temporary</b> - A temporary document not saved (DocumentType was SalesOrder, PurchaseOrder, ReturnOrder)  <b>Saved</b> - A saved document (DocumentType was SalesInvoice, PurchaseInvoice, ReturnInvoice) ready to be posted.  <b>Posted</b> - A posted document (not committed). See <a href="#">PostTax</a> for information on posting a document.  <b>Committed</b> - A posted document that has been committed. See <a href="#">CommitTax</a> for information on committing a document.  <b>Cancelled</b> - A committed document that has been cancelled. See <a href="#">CancelTax</a> for information on cancelling a document.  <b>Adjusted</b> - A document that has been adjusted through See <a href="#">AdjustTax</a> call.	Output



Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
					<b>Any</b> - Any status (used for searching)	
Reconciled	5	Char			True if the document has been reconciled. Only committed documents can be reconciled.	Output
TimeStamp	30	Char			Date of the last status change on the document (that is, Save date, Post date, Commit date, Cancel date). This will get converted in LocalTimeZone.	Output
TotalAmt	15	Dec	6		The sum of all line Amount values.	Output
TotalDisc	15	Dec	6		The sum of all TaxLine Discount amounts. Typically it will equal the requested Discount, but, it is possible that no lines were marked as discounted.	Output
TotalExmpt	15	Dec	6		The sum of all TaxLine Exemption amounts.	Output
TotTaxable	15	Dec	6		The amount the tax is based on. This is the total of all Line Base amounts. Typically it will be equal to the document	Output
TotalTax	15	Dec	6		The total tax for the document.	Output
TotTaxCalc	15	Dec	6		TotalTaxCalculated indicates the total tax calculated by AvaTax. This is usually the same as the TotalTax, except when a tax override amount is specified. This is for informational purposes. The TotalTax will still be used for reporting.	Output
HashCode	10	Int	0		Hash Code.	Output
Locked	5	Char			Flag indicating if a Document has been locked by Avalara's Managed Returns Service (MRS) process. Locked documents cannot be modified and cannot be cancelled because they have been reported on Tax Return.	Output
AdjustRsn	3	Int	0		Returned by the AdjustTax Service.  Sets a valid reason for the given AdjustTax call. Adjustment Reason is a high level classification of why an Original Document is being modified.  0 - Not Adjusted 1 - Sourcing Issue 2 - Reconciled with General Ledger 3 - Exemption Certificate Applied 4 - Price or Quantity Adjusted 5 - Item Returned 6 - Item Exchanged 7 - Bad Debt	Output

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
					8 - Other (Explain - Must provide AdjustmentDescription)	
AdjustDesc	50	Char			Sets description for the given AdjustTax call. Adjustment Description is required when AdjustmentReason is "Other" for enhanced tractability.	Output
Version	3	Int	0		Version of the web service (for example, 4.0.0.0)	Output
TaxDate	10	Char			Tax Date is the date used to calculate tax on the Document.	Output
Volatile	5	Char			Volatile Tax Rate (true or false)	Output
A Line returns the following parameters:						
No	10	Char		Yes	Line #	Output
TaxCod	10	Char		Yes	System or Custom Tax Code.	Output
Taxability	10	Char		Yes	Indicates that the taxability of the address should be checked.	Output
BoundLvl	10	Char		Yes	The level of jurisdiction boundary precision used for the tax calculation.	Output
Exemption	15	Dec	6	Yes	The portion of the detail that is exempt from taxes.	Output
Discount	15	Dec	6	Yes	The discount amount to apply to the document.	Output
Taxable	15	Dec	6	Yes	The taxable amount for the jurisdiction (or part of a jurisdiction in cases of tiered tax rates).	Output
Rate	15	Dec	6	Yes	The tax rate, that is the rate of taxation (0.0 - 1.0)	Output
Tax	15	Dec	6	Yes	The tax amount, that is the calculated tax (Taxable * Rate)	Output
TaxCalcd	15	Dec	6	Yes	TaxCalculated indicates the tax calculated by AvaTax. It will usually be the same as the Tax property, unless it has been overridden by a TaxAmount.	Output
TaxIncl	5	Char		Yes	TaxIncluded indicates the tax amount has already been included in the line Amount. The tax will be reversed out and returned in the Tax property. The Taxable property will indicate the calculated tax base amount.	Output
ExptCertId	10	Int		Yes	The Exemption Certificate number, if used.	

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
TaxDat	10	Char		Yes	Tax Date is the date used to calculate tax on the Document.	
RepDat	10	Char		Yes	Reporting Date	
AcctMethod	10	Char		Yes	Accounting method. 'Accrual' or 'Cash'	
An address returns the following parameters:						
Address	50	Char		Yes	Street address	Output
AddrCode	3	Char		Yes	Programmatically determined value used internally by the adapter. Defaults to the hash code of the Address object.	Output
BoundLevel	3	Int		Yes	The level of jurisdiction boundary precision used for the tax calculation.	Output
City	50	Char		Yes	City	Output
Country	2	Char		Yes	Country code (e.g. US)	Output
PostCode	10	Char		Yes	Postal or ZIP Code.	Output
Region	2	Char		Yes	Province or state.	Output
TaxRegId	10	Int		Yes	TaxRegionId provides the ability to override the tax region assignment for an address.	Output
Latitude	10	Char		Yes	Latitude	Output
Longitude	10	Char		Yes	Longitude	Output
GeoCodeTyp	20	Char		Yes	GeoCode Type (for example, StreetLevel)	Output
ValidSt	20	Char		Yes	Validation status (for example, NormalHit)	Output
DistToBnd	10	Int		Yes	Distance to boundary.	Output
Error Messages return the following parameters:						
Summary	50	Char		Yes	Gives the concise summary of the message.	Output
Details	50	Char		Yes	Gives more specific information about the nature of the exception.	Output
HelpLink	50	Char		Yes	Gives the URL to help page for this message.	Output

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
RefersTo	50	Char		Yes	Gives the name of a particular field if the service specified one to which this exception applies.	Output
Severity	50	Char		Yes	Gives the Severity Level of the message.	Output
Source	50	Char		Yes	Gives the source of the message.	Output

## Appendix D: Tax Jurisdiction Data Parameters

Parameter Name	Length	Type	Dec	Array	Description	I/O Usage
Country	2	Char		Dim(99,10) or dim(999,10)	Country name	Output
RegionID	10	Char		Ditto	State or Province	Output
JurisType	10	Char		Ditto	Jurisdiction Type (for example, State or County etc.)	Output
JurisCode	10	Char		Ditto	Jurisdiction Code for the taxing jurisdiction	Output
TaxType	10	Char		Ditto	Tax Type  S - Sales Tax U - Use Tax	Output
Base	15	Dec	6	Ditto	The tax base, that is the taxable amount.	Output
Taxable	15	Dec	6	Ditto	The taxable amount for the jurisdiction (or part of a jurisdiction in cases of tiered tax rates).	Output
Rate	15	Dec	6	Ditto	The tax rate, that is the rate of taxation (0.0 - 1.0)	Output
Tax	15	Dec	6	Ditto	The tax amount, that is the calculated tax amount (Taxable * Rate)	Output
TaxCalcd	15	Dec	6	Ditto	TaxCalculated indicates the tax calculated by AvaTax. It will usually be the same as the Tax property, unless it has been overridden by a TaxAmount.	Output
NonTaxable	15	Dec	6	Ditto	The portion of the detail that is not subject to taxes.	Output
Exemption	15	Dec	6	Ditto	Exempt amount for this line.	Output
JurisName	25	Char		Ditto	The jurisdiction name.	Output
TaxName	25	Char		Ditto	It further defines the tax and jurisdiction.	Output
TaxAuthTyp	3	Int	0	Ditto	X12+ Tax Authority Type	Output
TaxGroup	10	Char		Ditto	TaxGroup indicates the group code when special grouping is enabled for particular client applications. For other clients, it will be empty.	Output
RateTyp	1	Char		Ditto	RateType indicates the tax rate type. These currently include:  F = Food (US) G = General (US) S = Standard (VAT) R = Reduced (VAT) Q = Super Reduced (VAT) A = Reduced Rate 1 (VAT) B = Reduced Rate 2 (VAT) Z = Zero Rate	Output
StateAssNo	4	Char		Ditto	StateAssignedNo indicates the identifier given to the location by the state.	Output



## 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)
```

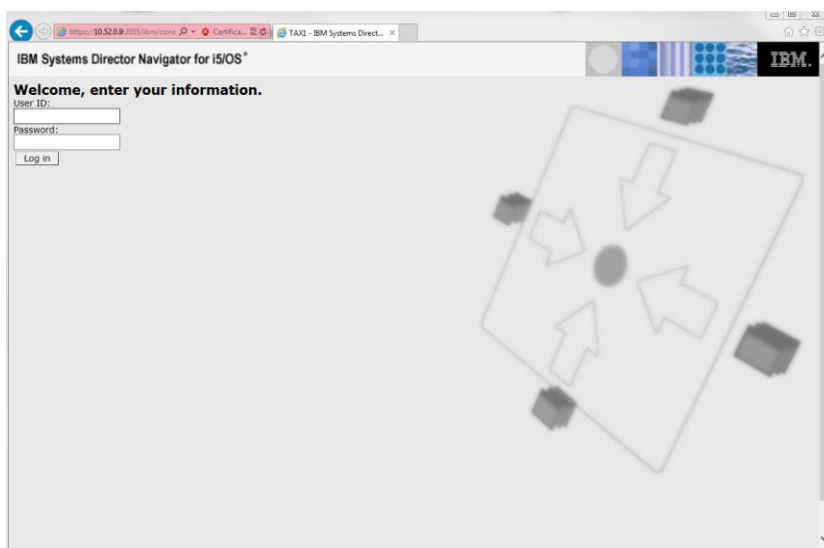
3. You can start the \*ADMIN server by issuing this command:

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

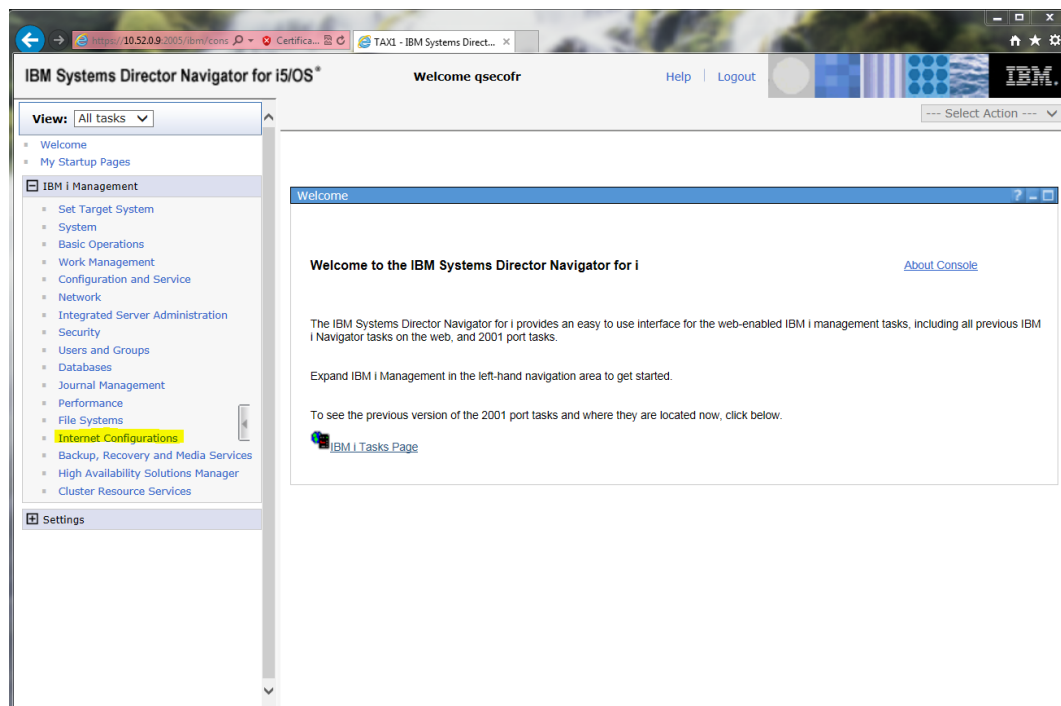
4. Open Windows Explorer and type <http://<iseriesipaddress>:2001> in the address bar. This should bring up the navigator sign on screen. Sign on as a security officer.

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

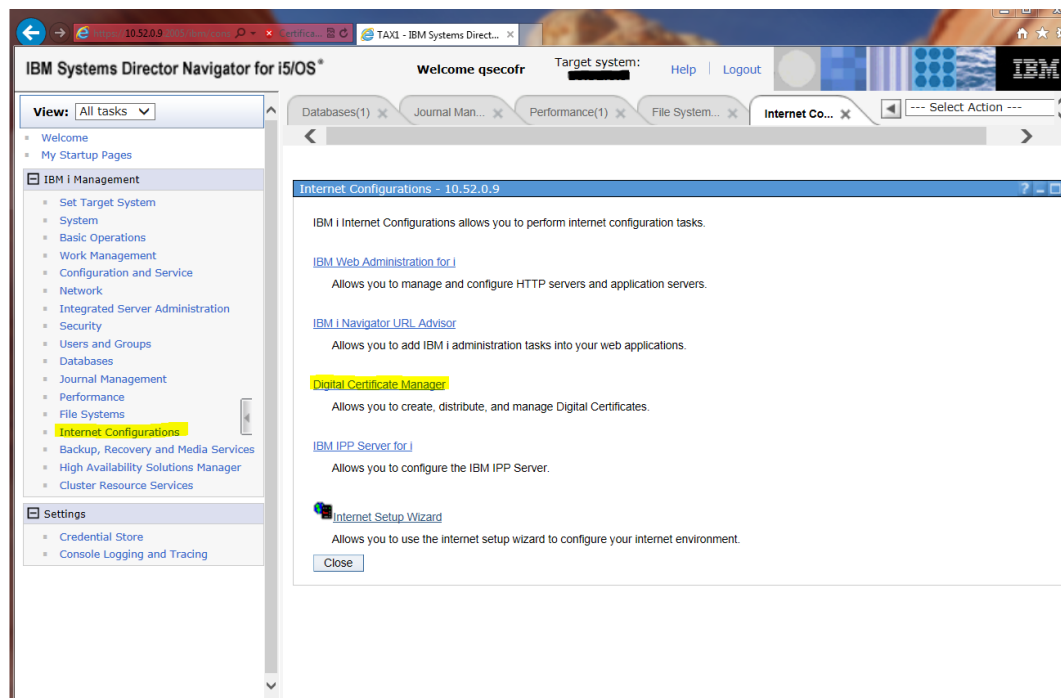
<http://<Your IBM IP Address>:2001/QIBM/ICSS/Cert/Admin/qycucm1.ndm/main0>



## Select internet configurations



## Select Digital Certificate Manager

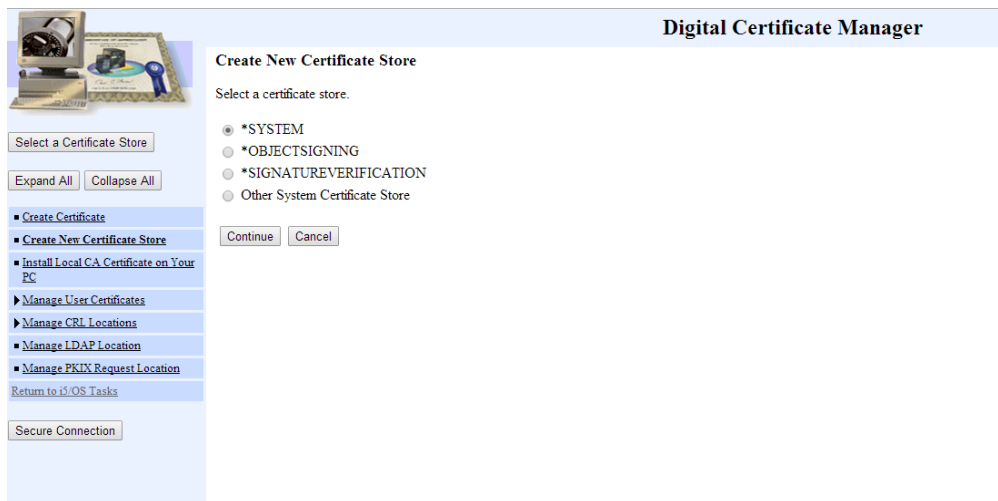




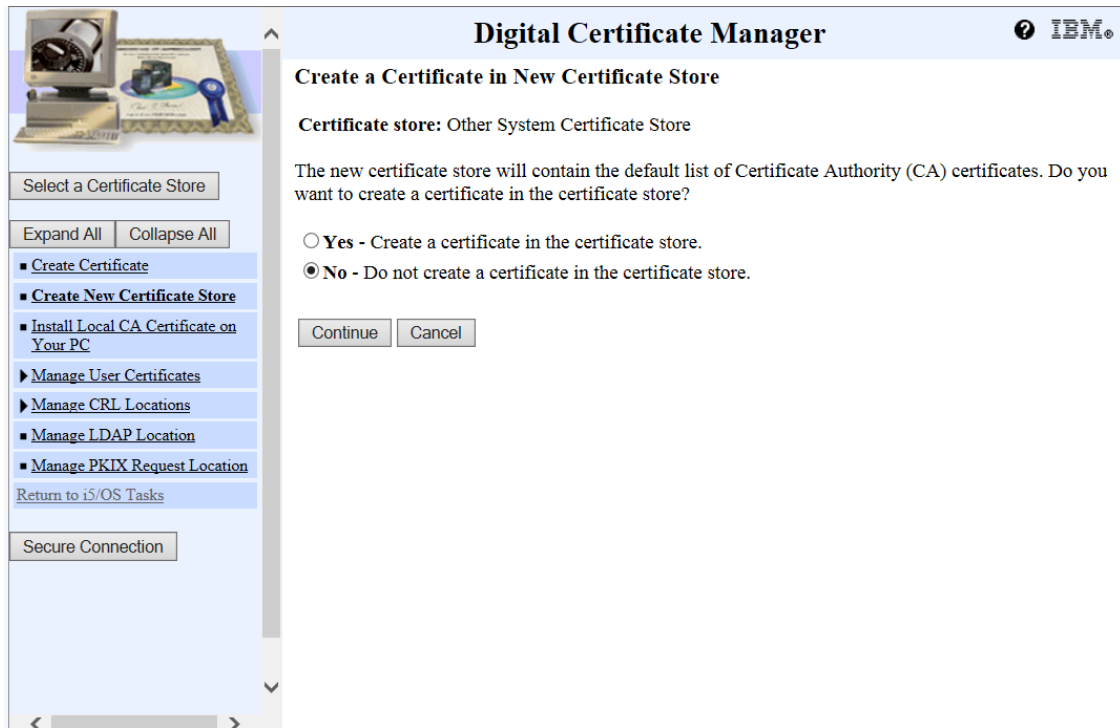
## Try to create a New Certificate store




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.



**Digital Certificate Manager** 

**Create a Certificate in New Certificate Store**

**Certificate store:** Other System Certificate Store

The new certificate store will contain the default list of Certificate Authority (CA) certificates. Do you want to create a certificate in the certificate store?

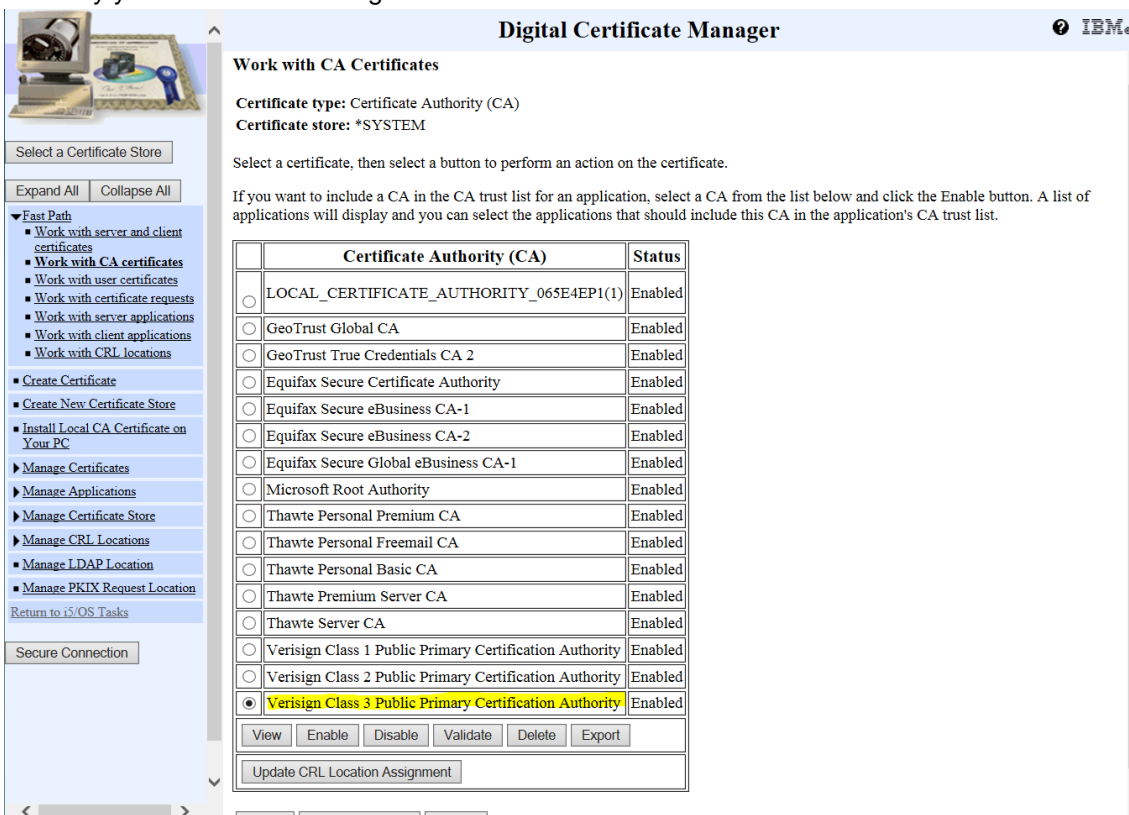
☐ **Yes** - Create a certificate in the certificate store.


☒ **No** - Do not create a certificate in the certificate store.

**Left Panel:**

- Select a Certificate Store
- Expand All Collapse All
- Create Certificate
- Create New Certificate Store
- Install Local CA Certificate on Your PC
- Manage User Certificates
- Manage CRL Locations
- Manage LDAP Location
- Manage PKIX Request Location
- Return to i5/OS Tasks
- Secure Connection

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.



**Digital Certificate Manager** 

**Work with CA Certificates**

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

Select a certificate, then select a button to perform an action on the certificate.

If you want to include a CA in the CA trust list for an application, select a CA from the list below and click the Enable button. A list of applications will display and you can select the applications that should include this CA in the application's CA trust list.

	Certificate Authority (CA)	Status
<input type="radio"/>	LOCAL_CERTIFICATE_AUTHORITY_065E4EP1(1)	Enabled
<input type="radio"/>	GeoTrust Global CA	Enabled
<input type="radio"/>	GeoTrust True Credentials CA 2	Enabled
<input type="radio"/>	Equifax Secure Certificate Authority	Enabled
<input type="radio"/>	Equifax Secure eBusiness CA-1	Enabled
<input type="radio"/>	Equifax Secure eBusiness CA-2	Enabled
<input type="radio"/>	Equifax Secure Global eBusiness CA-1	Enabled
<input type="radio"/>	Microsoft Root Authority	Enabled
<input type="radio"/>	Thawte Personal Premium CA	Enabled
<input type="radio"/>	Thawte Personal Freemail CA	Enabled
<input type="radio"/>	Thawte Personal Basic CA	Enabled
<input type="radio"/>	Thawte Premium Server CA	Enabled
<input type="radio"/>	Thawte Server CA	Enabled
<input type="radio"/>	Verisign Class 1 Public Primary Certification Authority	Enabled
<input type="radio"/>	Verisign Class 2 Public Primary Certification Authority	Enabled
<input checked="" type="radio"/>	Verisign Class 3 Public Primary Certification Authority	Enabled

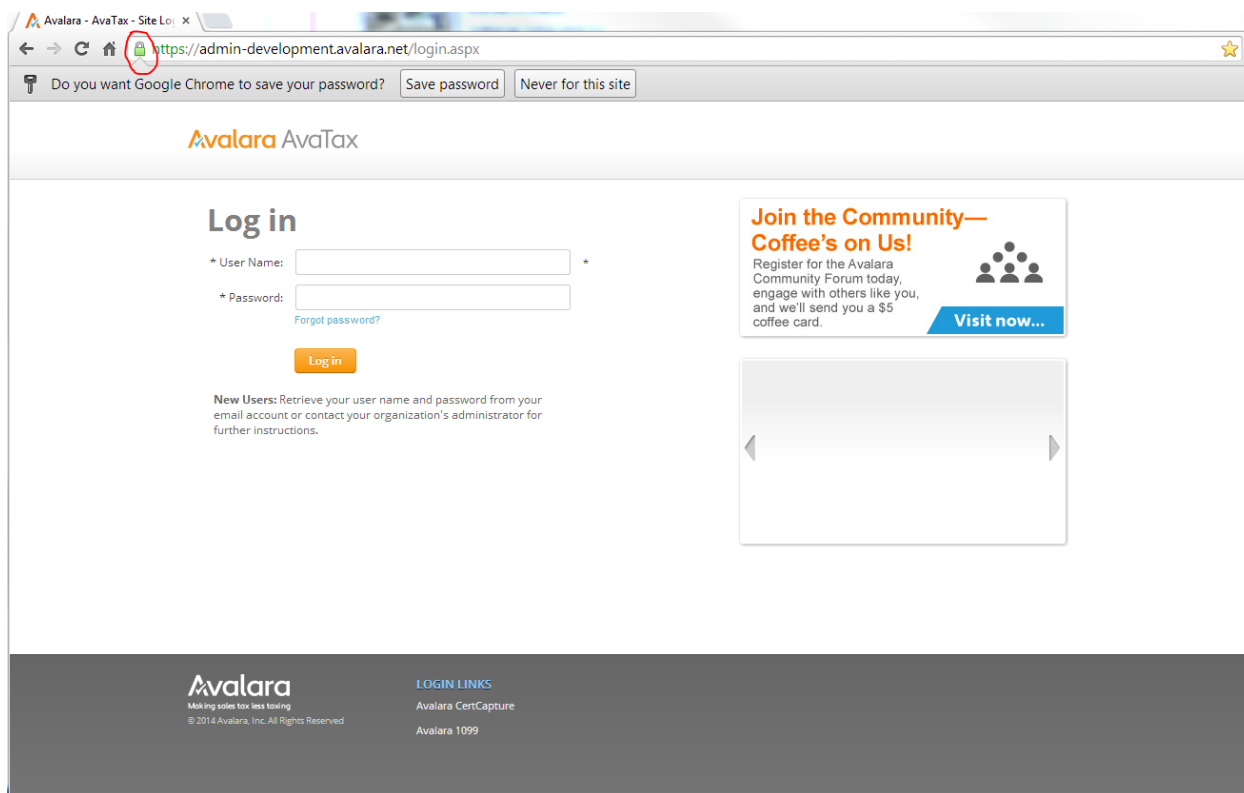
**Left Panel:**

- Select a Certificate Store
- Expand All Collapse All
- Fast Path
  - Work with server and client certificates
  - Work with CA certificates
  - Work with user certificates
  - Work with certificate requests
  - Work with server applications
  - Work with client applications
  - Work with CRL locations
- Create Certificate
- Create New Certificate Store
- Install Local CA Certificate on Your PC
- Manage Certificates
- Manage Applications
- Manage Certificate Store
- Manage CRL Locations
- Manage LDAP Location
- Manage PKIX Request Location
- Return to i5/OS Tasks
- Secure Connection

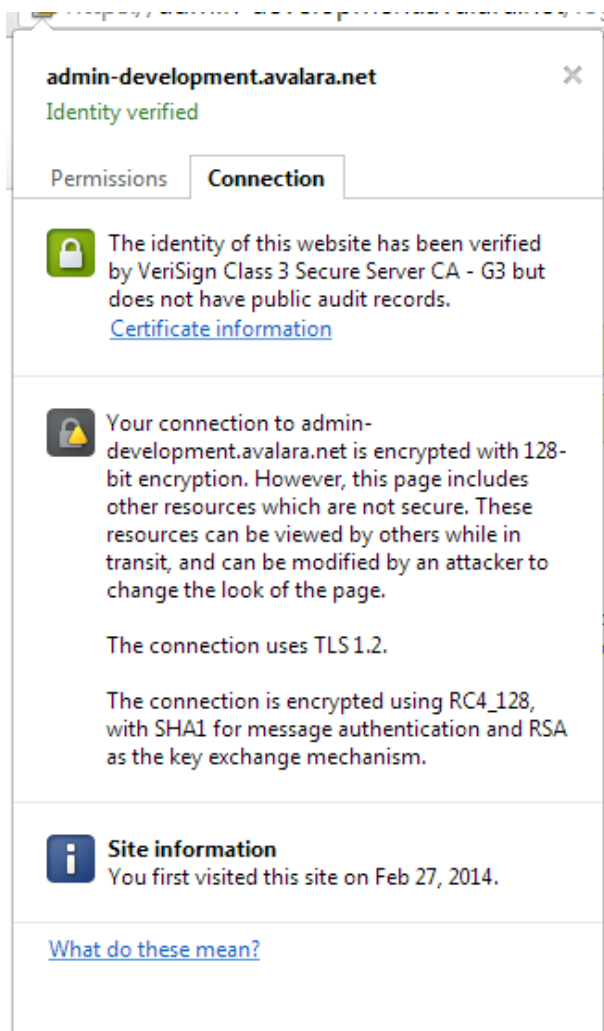
That should be all you need to do for setting up the certificate store apart from ensuring that your users have authorization access.

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.

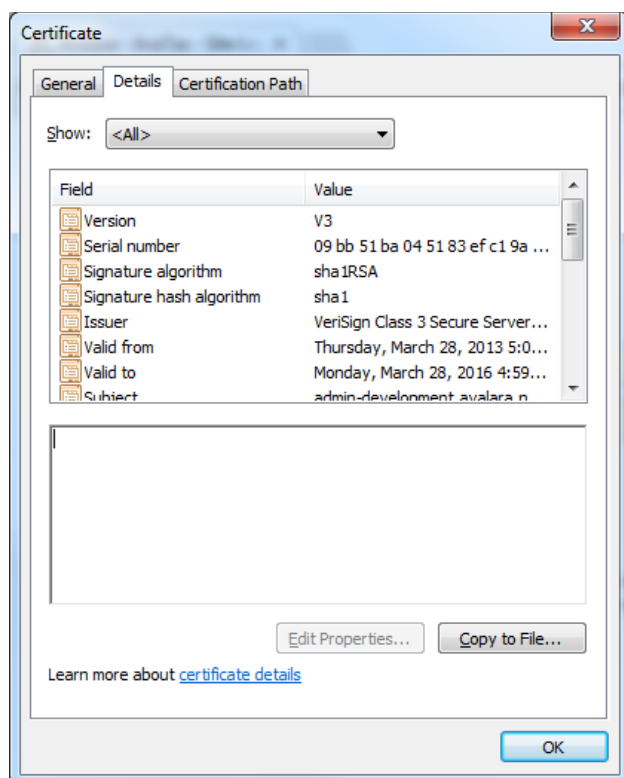
Go to our development console login page. Note the green padlock in the address bar.



Right click on the padlock and select the connection tab to display the drop down below. Click on [Certificate information](#).



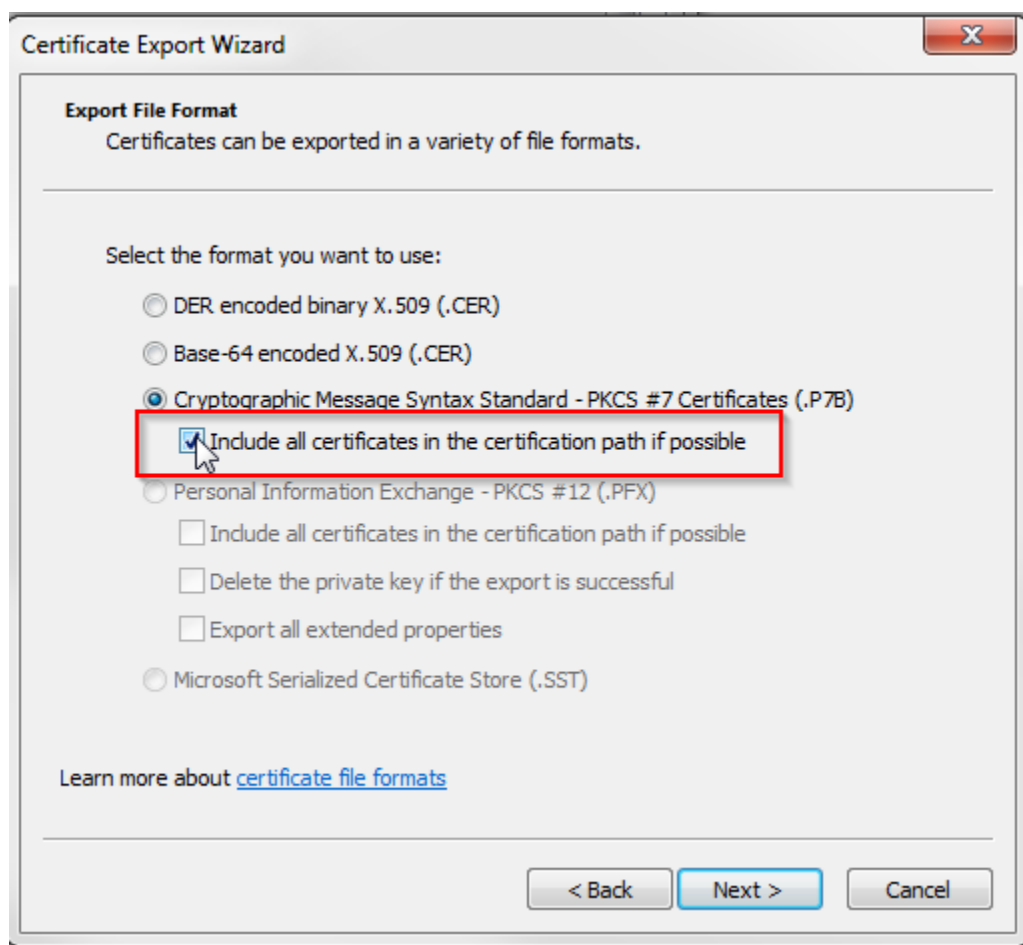
Click on the Details tab and then the 'Copy to File' button. This will start the Certificate Export Wizard.



Click Next >

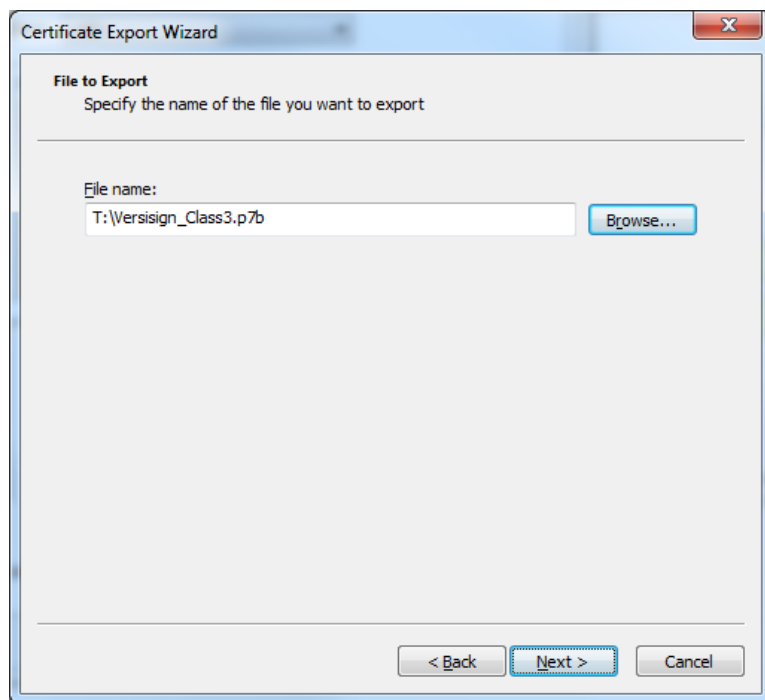


Select the .P7B format and Include all. Hit Next >

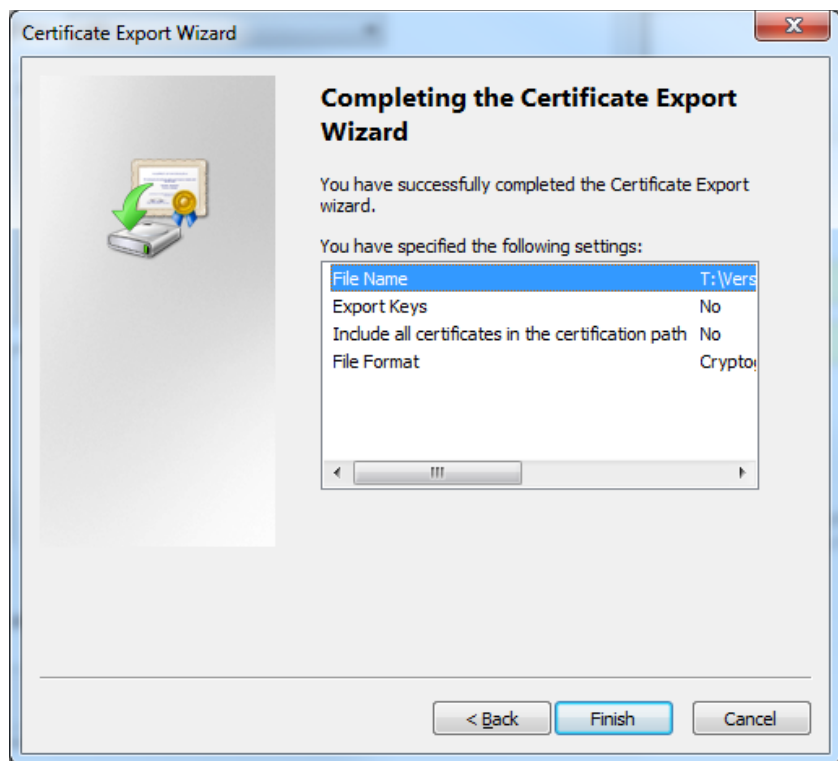


Give the file a location and save name (we recommend saving in the /tmp directory of your IFS).

Hit next >

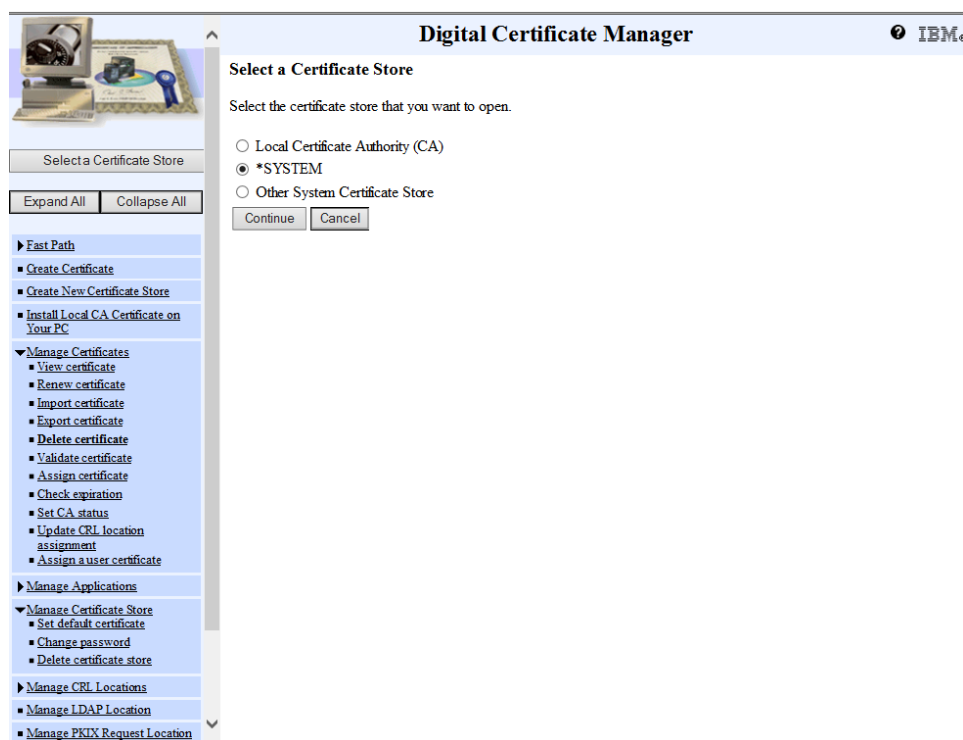


Hit Finish and you should get an 'Export Successful' message.

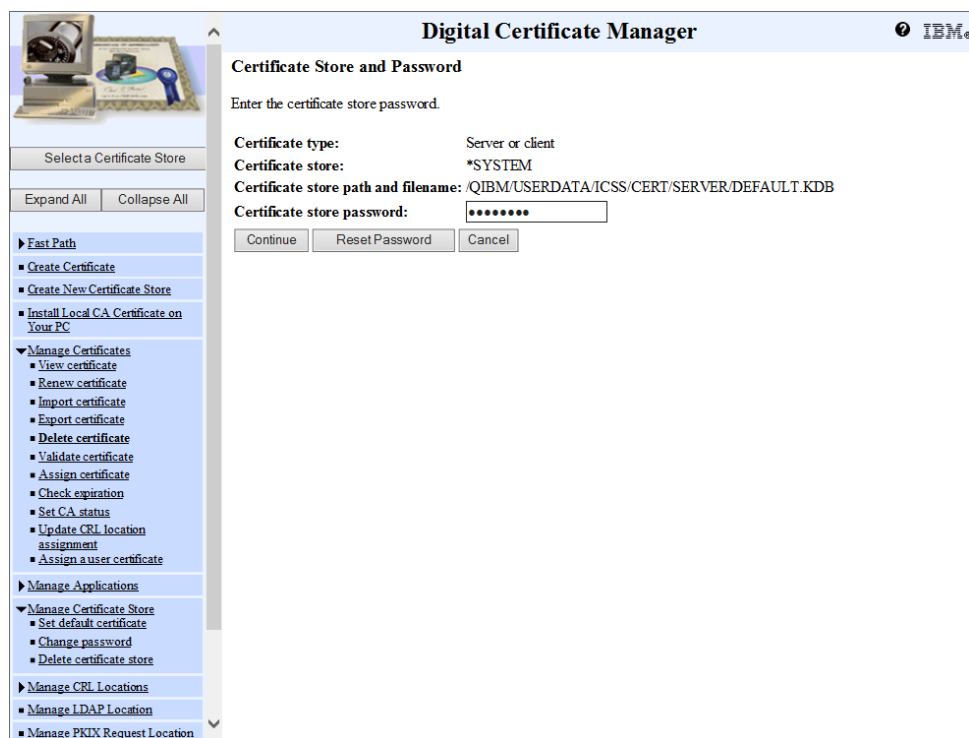




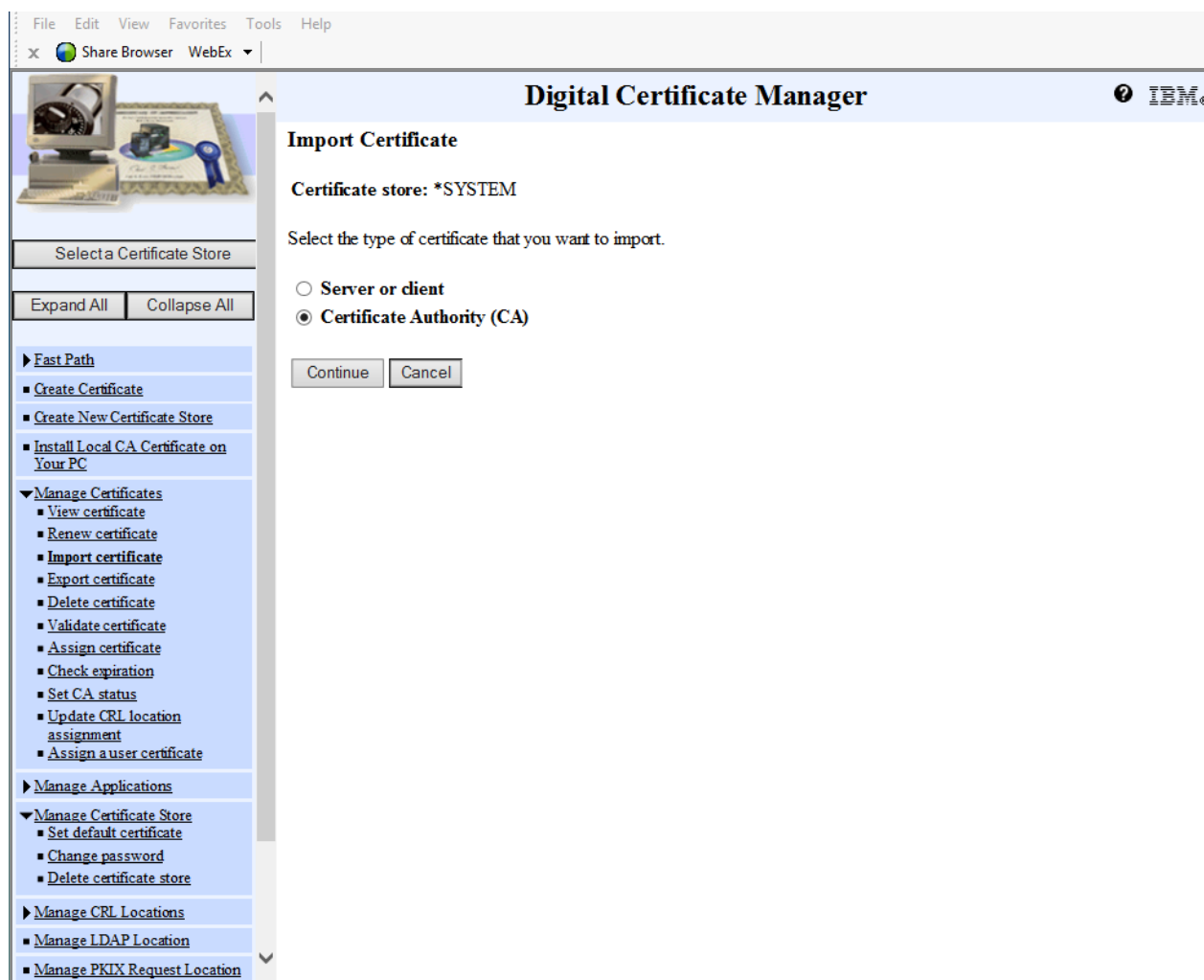
Now go back to DCM and select the \*SYSTEM store



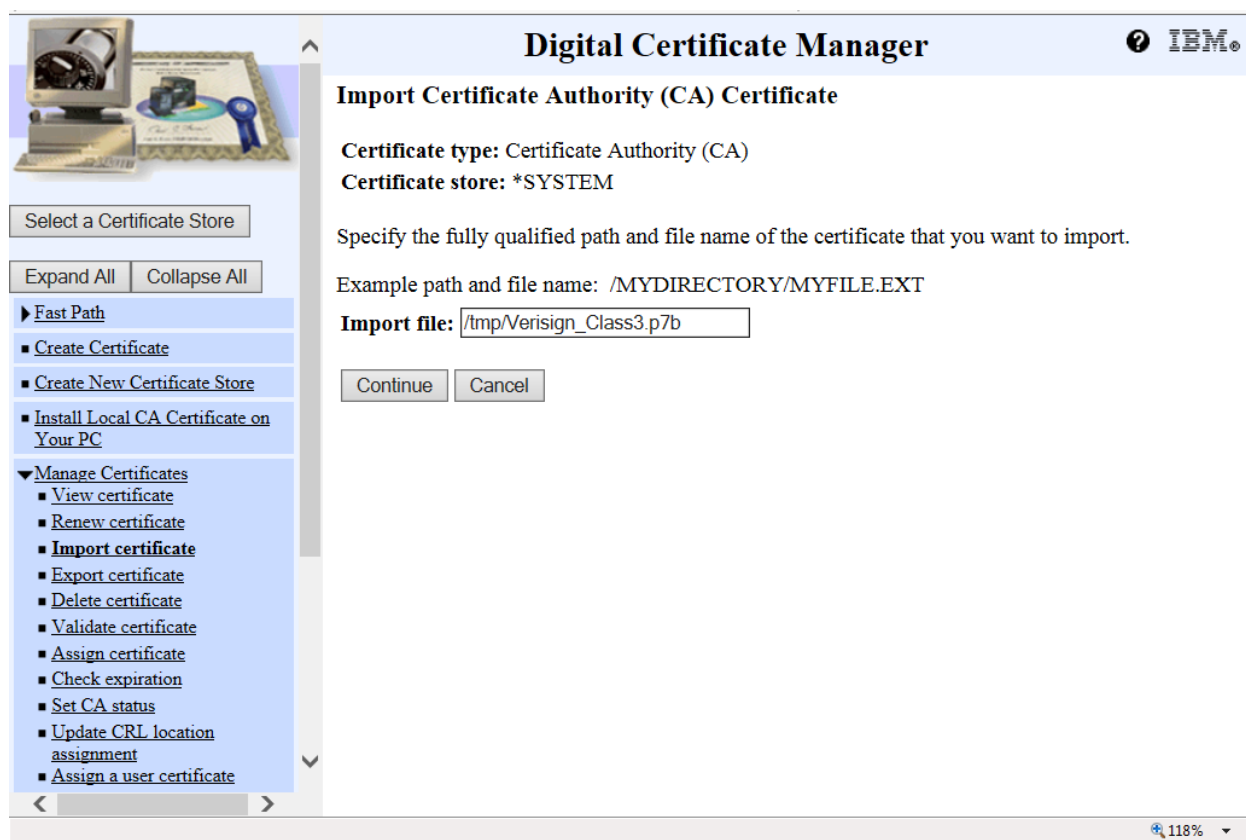
Enter your password



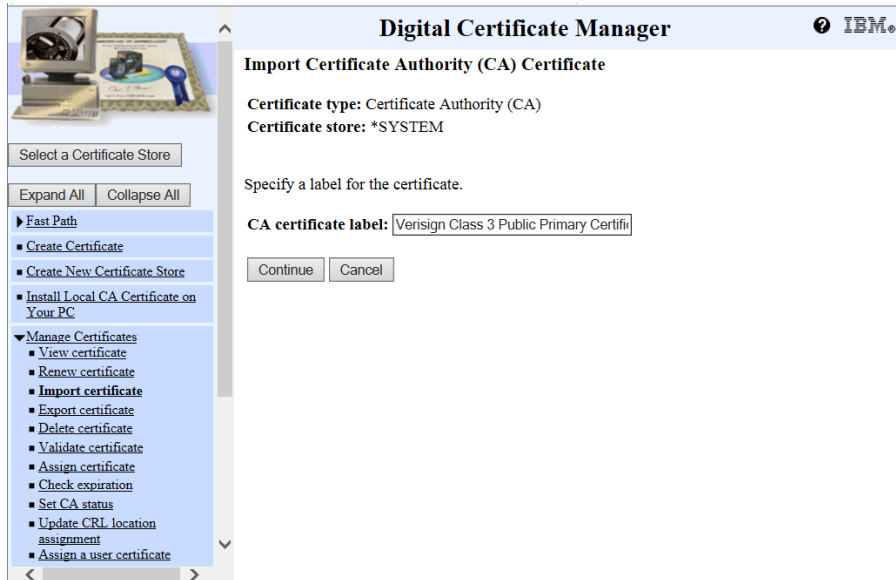
## Select Import Certificate > Certificate Authority (CA)



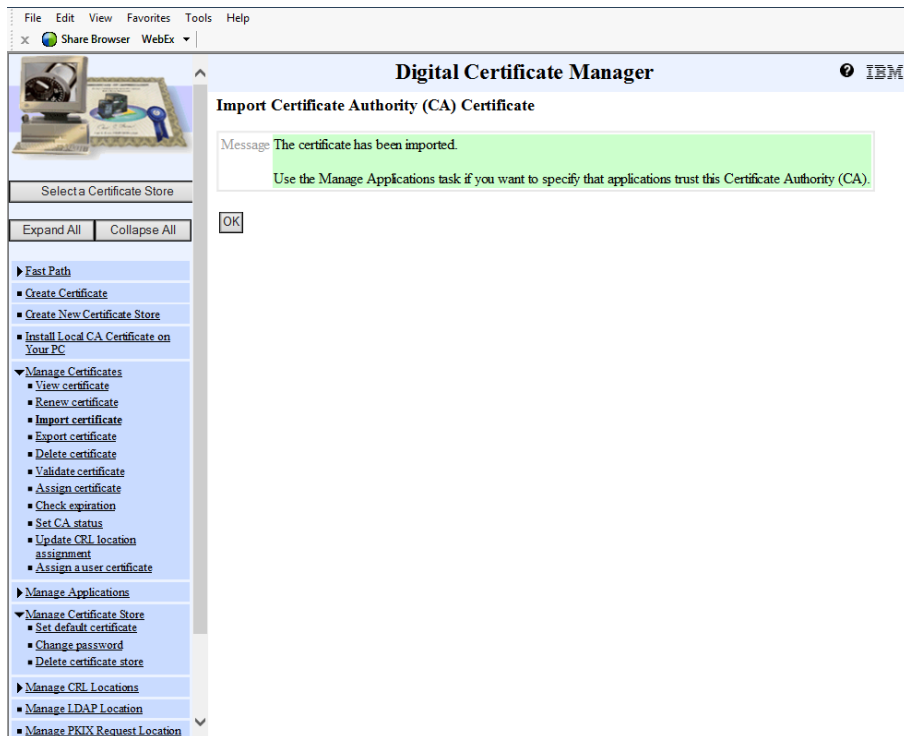
Enter the location and name of the import file that you saved above.



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 adapter is included with your install library.

The tax service adapter can make an optional call to the address service before calling the tax service. In this case the tax service can use the address returned from the address service to potentially make a more accurate tax call. This is selected by setting the Address Validation flag in the System Control file.

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 AX005R1 as an example. This program can be seen running from the Inquiry menu.

AX008R1 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 application look at the source in AVATAX/QRPGLESRC AX008R1 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 ISO 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.

```

AC0004R1                               Maintain Entity Control File                               11/04/14
                                                                                               15:12:33

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

   US  _  <= Position
Opt ISO3 Typ ISO2 Entity Name                      Cur Addr
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.

## Appendix H: Change Log

Release	Date	Changes
13.2.4	July 2014	<ol style="list-style-type: none"> <li>1) Fixed bug in address validation. The on/off flag at the System Control level for GetTax and AdjustTax was being ignored.</li> <li>2) Removed the address validation on/off flag from those services where it was not meaningful. Now only applies to GetTax, AdjustTax and Validate.</li> <li>3) Improved granularity of address checking that is controlled by the (optional) Entity Control file. It can now be managed at a country, state or province level.</li> <li>4) Simplified the switching between Dev and Prod accounts in the System Control file. This was previously done by changing a URI string. It now handled by a simple switch.</li> <li>5) New auto-installer has been built in that upgrades SDK versions while keeping customer setup data intact.</li> </ol>
13.2.5	Aug 2014	<ol style="list-style-type: none"> <li>1) Added support for GetTaxHistory service</li> <li>2) Clear the AXO output data structure before calling the service</li> </ol>
13.2.6	October 2014	<ol style="list-style-type: none"> <li>1) Business Identification No increased to 25 bytes long</li> <li>2) Skeleton Address Cleanser included</li> <li>3) Minor fixes to System Control file displays for Address Service and validation.</li> </ol>
13.2.7	November 2014	<ol style="list-style-type: none"> <li>1) Adapter now accepts ISO-3 country codes and automatically converts to ISO-2 using the AX0004P Entity Control file. This provides a simple table driven method for handling 2 or 3 character country codes. The input country code field has been increased in size from 2 to 3 bytes.</li> <li>2) Subfile position functionality has been added to the Entity Control Maintenance program AC004R1.</li> </ol>
13.2.8	December 2014	<ol style="list-style-type: none"> <li>1) Minor changes to the profile settings to provide more consistency.</li> </ol>
14.1.0	February 2015	<ol style="list-style-type: none"> <li>1) Simplified data structure use. /Copy member now provides all the data structure definitions required to use the services.</li> <li>2) Removed I_ and O_ pre-fixes from data structure field names for Tax Services.</li> <li>3) Removed U_ and V_ pre-fixes from data structure field names for Address Service.</li> <li>4) Simplified the System Control interface.</li> </ol> <p><b>Note:</b> This version is not directly compatible with connector code developed under previous versions. See Appendix J – Upgrading from release 13 to 14.</p>
14.1.1	June 2015	<ol style="list-style-type: none"> <li>1) Fixed a bug that occurred when using Address Validation with GetTax.</li> </ol>

14.1.2	June 2015	1) Number of allowed order/invoice detail lines increased to 4500 and number of addresses increased to 150 for the V6R1 and newer version of the SDK adapter.
14.1.3	July 2015	Some numeric fields were defined as size 15 6. This caused a field size issue within the adapter when using very large numbers (like 1000, 000, 000). Floating point field sizes Field sizes have now been defined as 15 4.
14.1.4	November 2015	The following characters which were causing XML or translation issues in previous releases are now handled by the adapter without error.  ÀÂÃÄÅËÊËÏÔÕÙàâãäåêëïòûüÿÇç«»&<>°çμ
14.1.5	March 2016	1) *Auto may be used for the debug filename. This will generate uniquely named logs that will not be overwritten. See the section <b>Configure the System Control File</b> for more details.  2) The SDK adapter now has support for the Avalara CloudConnect service.
14.1.6	September 2016	1) Added feature to the F1 Info button on the System Control that displays the latest release available. 2) Fixed bug where the adapter version was showing incorrectly in the System Control display. 3) Extended XML character 'cleansing' to include line level Exemption Number field.
14.1.7	October 2016	Customer with QSSLPCL set to *TLSV1.2 *TLSV1.2 reported the following error.  (GSKit) Operation is not supported by SSL. ssl_error(412): (GSKit) Operation is not supported by SSL. SetError() #24: gsk_env_init: (GSKit) Operation is not supported by SSL.  This release corrects that error.
14.1.8	February 2017	1) Added Customer Code to the list of fields that are checked for characters known to cause XML or translation issues. The full list of fields that are checked are: a) Document Code b) Salesperson c) Purchase Order d) Customer Code e) Exempt No. f) Reason g) Currency h) All Addresses



		2) Included the addition of a program that can return the tax breakdown for Canadian orders/invoices. For some customers this is required information to be printed on their invoice. The program is called AX009R1. More details can found in this document.
14.1.9	June 2017	Corrected a situation where the SDK adapter was only processing the first 150 addresses. It now handles 450.
14.1.10	July 2017	<p>Added the ability for a customer to modify the web service timeout value. By default, this is set to 180 seconds (i.e. if the SDK adapter does not receive a response within that time then it drops the connection). The timeout value can be modified by changing data area AVATAX/AC0009DA to contain the number of seconds that you wish the adapter to wait for a response.</p> <p>For example, CHGDTAARA DTAARA(AVATAX/AC0009DA) VALUE(300) would set the timeout for 300 seconds (5 minutes).</p>
14.1.11	Sept 2017	Certain characters were causing the SDK parser to fail with a 'HTTP/1.1 500 Internal Server Error'. This release addresses that problem.
14.1.12	Oct 2017	<p>Corrects an issue where the AXO.TimeStamp value was not being returned to the caller.</p> <p>Increased the size of AXI.RefCode from 10 to 50</p>
14.1.13	Dec 2017	<p>Removed the fix put in at 14.1.4. The 14.1.11 change handled this in a more efficient way and this conversion code is no longer needed.</p> <p>XML Illegal Characters: The adapter now automatically replaces '&amp;' '&lt;' and '&gt;' with their appropriate XML character sequences so that they do not create an error condition. In the Admin Console, any fields containing these characters will appear just as they were sent.</p> <p>Any customers wishing to handle these characters themselves can do so by setting the data area, AC0010DA = 'N'. This will switch off the automated handling of the illegal characters.</p> <p>It is shipped with the default value of 'Y'. We recommend that you keep it that way.</p>
14.1.14	April 2018	Receiver size for the DistanceToBoundary output field was too small for users using Lat/Long. Increased size to 10.0 integer.
14.1.15	Nov 2018	Avatax has started using Doc ID's greater than 10 digits. Increased the size of DocId to 25. Any customer using this field in their connector code should also resize to 25.

## Appendix J – Upgrading from release 13 to 14

Release 14, starting with version 14.1.0, cannot be automatically upgraded from release 13. This appendix explains the process for moving between releases 13 to 14.

The AVASTALLER available from our FTP site (described in the section headed ‘Installing the AvaTax SDK for IBM i’) now performs as follows:

- 1) If this is a new install (i.e. the Avalara library you are installing does not exist on your system) then a fresh copy of the latest version 14 will be installed.
- 2) If this is an upgrade (i.e. the Avalara library you are installing already exists on your system) then;
  - a) If you are already on a version 14 release you will be upgraded to the latest version of that release.
  - b) If you are already on version 13 then you cannot upgrade automatically to version 14 and should follow the instructions in the rest of this appendix.

### Release 13 to 14 Upgrade Instructions

**These instructions assume that your release 13 library is called AVATAX and your temporary release 14 library will be called AVATAX2**

- 1) Remove AVATAX from your library list.

```
RMVLIBLE LIB(AVATAX)
```

- 2) Install a fresh copy of release 14 on your system by following the instructions under the section in this manual titled ‘Installing the AvaTax SDK for IBM i’ to. Use steps 1,2,4,5 and 6.

IMPORTANT: For step 5 make sure that the ‘Library to Install or Upgrade’ does not exist on your system. So, for example, if you previously installed release 13 into library AVATAX then install 14 into AVATAX2.

- 3) Now you will need to copy your custom settings from your release 13 library to your new release 14 library. You can cut and paste the CPYF commands below.

```
CPYF FROMFILE(AVATAX/AX0001P) TOFILE(AVATAX2/AX0001P) MBROPT(*REPLACE)
CRTFILE(*NO) FMTOPT(*MAP *DROP)
```

```
CPYF FROMFILE(AVATAX/AX0002P) TOFILE(AVATAX2/AX0002P) MBROPT(*REPLACE)
CRTFILE(*NO) FMTOPT(*MAP *DROP)
```

```
CPYF FROMFILE(AVATAX/AX0003P) TOFILE(AVATAX2/AX0003P) MBROPT(*REPLACE)
CRTFILE(*NO) FMTOPT(*MAP *DROP)
```

```
CPYF FROMFILE(AVATAX/AX0004P) TOFILE(AVATAX2/AX0004P) MBROPT(*REPLACE)
CRTFILE(*NO) FMTOPT(*MAP *DROP)
```

```
CPYF FROMFILE(AVATAX/SC0005P) TOFILE(AVATAX2/SC0005P) MBROPT(*REPLACE)
CRTFILE(*NO) FMTOPT(*MAP *DROP)
```

- 4) Finally you will need to make the following changes to any connector program you have written that calls the **Avalara tax or address adapters**.

- a) Backup your original source members or make copies.
- b) Remove this data structure from your connector. It is now provided by the /Copy member.

```
D AmountC      S          10H          121115
//                                     130206
// For jurisdiction breakdown          130206
//                                     130206
D LineCnt      S          3i 0          121119
D i1           S          3i 0          121119
/ Add these data structure definitions in the CALLING program 130207
D HTI          DS          likeds(HttpI) 130207
D AXI          DS          likeds(AsiaI)  130207
D AX0          DS          likeds(Asia0)  130207
D JUO          DS          likeds(TxJu0)  130207
D*             130208
D* Definitions 130208
D*             130211
D* Constants   130211
D*             130211
D @Success     c           'Success'      130327
D @Err         c           'Error'        130211
D @NoService   c           'No active tax service' 130211
D*             130208
```

- c) Replace any reference to TxJuO with JUO. Below are examples of where you might find reference to TxJuO.

```
AXI.I_Amt(3) = Amt3;          130328
AXI.I_Disc(3) = 'false';      130328
if Disc3 = 'Y';               130401
    AXI.I_Disc(3) = 'true';    130328
endif;                         130328
endif;                         130328
// Call tax service           130328
AvataxWebSvc (HTI:AXI:AX0:TxJu0); 130328
// Check the Result Code for errors 130212
if AX0.0_StatusCode <> @Success; 130328
```

```

// Check if user hit exit                                     130226
if *in03 = *off;                                             130226
// If not exit the clear buffers                             130226
clear HttpI;                                                130226
clear AsiaI;                                                130226
clear Asia0;                                                 130226
clear TxJu0;                                                 130226
// Get an active GetTax header to use for the request        130226
chain '001' ax000111;                                       130226

rrn1 = rrn1 + 1;                                             130211
do (TxJu0(1).O_TaxName(rrn1)=*Blanks) or (rrn1 >= 10);    130225
  TaxName1 = TxJu0(1).O_TaxName(rrn1);                     130208
  TaxName2 = TxJu0(1).O_JurisName(rrn1);                   130211
  TaxRate = TxJu0(1).O_Rate(rrn1)*100;                     130208
  eval(H) TaxAmt = TxJu0(1).O_Tax(rrn1);                   130208
  write sfl1;                                               130211
  rrn1 = rrn1 + 1;                                          130211
enddo;                                                       130211
//                                                         130208

```

- d) For release 14 in order to improve readability the I\_ , O\_ , V\_ and U\_ prefixes were removed from the data structure field names.

The easiest way to correct these is to use the Scan/Replace function available in SEU. Here is an example for replacing I\_ in a connector written in free format RPG.

F14 will bring up the Find/Change Options screen below. In Find put .I\_ in Replace put . Set allow data shift = 'Y'

Find/Change Options		
Type choices, press Enter.		
Find . . . . .	.I_	
Change . . . . .	.	
From column number . . . . .	1	1-100
To column number . . . . .	100	1-100 or blank
Occurrences to process . . . . .	1	1=Next, 2=All, 3=Previous 4=First, 5=Last
Records to search . . . . .	1	1=All, 2=Excluded 3=Non-excluded
Kind of match . . . . .	2	1=Same case 2=Ignore case
Allow data shift . . . . .	Y	Y=Yes, N=No
Search for date . . . . .	15/02/25	YY/MM/DD or YYMMDD
Compare . . . . .	_	1=Less than 2=Equal to 3=Greater than
F3=Exit    F5=Refresh    F12=Cancel    F13=Change session defaults F15=Browse/Copy options    F16=Find    F17=Change		

Hit F16 to find and F17 to change. The result should look like this ...

```
Columns . . . : 6 100 Edit
SEU==>
FMT ** ... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
0124.00 // 130328
0125.00 AXI.Company = Company; 150225
0126.00 if DocId <> *Blanks; 130401
0127.00 AXI.DocId = DocId; 150225
0128.00 endif; 130401
0129.00 AXI.DocCode = DocCode; 150225
0130.00 AXI.NewDocCode = newcode; 150225
0131.00 AXI.DocType = DocType; 150225
0132.00 // Set to current date 130328
0133.00 AXI.DocDate = %char(DocDate); 150225
0134.00 AXI.CusCode = Customer; 150225
0135.00 AXI.I_ExemptNo = ExemptNo; 130328
0136.00 AXI.I_PurchOrd = PONumbr; 130328
0137.00 AXI.I_SalesPsn = SalesP; 130328
0138.00 AXI.I_Discount = DiscAmt; 130328
0138.01 AXI.I_CusUseTy = CustUse; 141021
0139.00 AXI.I_LocCode = LocCde; 130328
0140.00 AXI.I_OrigCode = '123'; 130328
0141.00 AXI.I_DestCode = '456'; 130328
0142.00 AXI.I_DetailLvl = 'Tax'; 130328

F3=Exit F4=Prompt F5=Refresh F9=Retrieve F10=Cursor F11=Toggle
F16=Repeat find F17=Repeat change F24=More keys
Change processed.
```

AXI.I\_Company is changed to AXI.Company etc.

Repeat the process for all your own tax and address connectors to replace the I\_ , O\_ , V\_ and U\_ prefixes

Finally make sure AVATAX2 is in your library list and recompile the connectors you have changed.

## 5) Test your release 14 setup.

Make sure you are using a company that is attached to your sandbox account.

a) With AVATAX2 in your library list type ASIA to bring up the menu system.

Then select 1) Maintenance > 1. Maintain System Control > Enter 'T' against your GetTax service and hit <Enter>. You should see a screen similar to the one below with a successful connection.

```

AC001R1                               Maintain System Control File                               2/25/15
                                                                                               11:43:24

Type option, press Enter .....
 2=Edit 3=Copy 4=Delet :      Test Connection Settings Result      :
:                                                                    :
Opt Primary Service :      Result: Success      :
- Tax Services      :                                                                    :
- Tax Services      :      Tran. ID: 1015240162      :
I Tax Services      :                                                                    :
- Tax Services      :      Version: 15.1.1.1      :
- Tax Services      :                                                                    :
- Tax Services      :      F12=Cancel      :
- Tax Services      :                                                                    :
- Tax Services      : .....:
- Tax Services      Ping      1
- Address Services  Validate      1
- Address Services  Ping      1

F1=Info  F3=Exit  F6=Add
Copyright 2004-2014 Avalara, Inc. All Rights Reserved

```

- b) Next use 21. Integration Tester to perform a **SalesOrder** and check that you are getting tax calculated.
  
- 6) Finally, once you are happy that your new version is working OK you can rename your AVATAX library to AVATAX\_OLD and AVATAX2 to be AVATAX. Then make sure you rename you old connectors and move your newly compiled connectors into the library you use to store them.

## 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](mailto:taxrates.support@avalara.com) or via chat at <http://www.avalara.com/Technical-Support>.