

# Infor FACTS Sales Tax

Powered by AvaTax™

## AvaTax Connector

FACTS 7.5 – 9.3

User's Guide

Version 9.3.L – April 2024

*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.*

© 2022 Avalara, Inc. All rights reserved.

© 2022 Sequoia Group, Inc. All rights reserved.

*Avalara, AvaTax Calc, AvaTax Returns, Exemption Certificate Management, and AvaRates 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

Table of Figures.....	4
INTRODUCTION.....	6
ACTIVATING YOUR AVALARA ACCOUNT: AN OVERVIEW.....	6
THE AVALARA CONNECTOR AND YOUR ERP: AN OVERVIEW.....	7
The FACTS AvaTax Connector Installation Process.....	7
How the AvaTax Connector works within FACTS.....	8
CONNECTOR ADMINISTRATION.....	9
AvaTax Control F/M (TXF990).....	10
AvaTax Control F/M – Settings Tab Field Definitions.....	10
AvaTax Control F/M – Options Tab Field Definitions.....	11
AvaTax Control F/M – ECM Tab Field Definitions.....	13
AvaTax Control F/M – Communications Tab Field Definitions.....	15
AvaTax Control F/M – Basis Tab Field Definitions.....	16
AvaTax Control F/M – Statistics Tab information.....	16
Options By State F/M.....	17
Branch Address F/M Field Definitions.....	18
AvaTax Tax Code F/M Field Definitions.....	19
Obsolete Tax Codes.....	20
AvaTax Entity/Use Code F/M Field Definitions.....	20
Customer/Item Configuration.....	21
Exemption Certificate Update.....	23
Calculate Taxes Option Definitions.....	24
Commit Invoices to AvaTax Option Definitions.....	25

AvaTax History Removal Options Definitions.....26

Verified Addresses F/M Field Definitions.....26

Verify Addresses Options Definitions.....27

Address Verification Log Options Definitions.....28

Post Past Invoices.....29

Tax Detail Inquiry.....30

Sales Tax Adjustment Register.....31

THE CONNECTOR AND YOUR ERP.....33

    SO Static Control F/M Options.....33

    AR Invoice Processing Control .....34

    Customer File Maintenance Options.....35

    City Correction File Maintenance.....35

    Item File Maintenance Added Field Definitions.....36

    Item Class File Maintenance Field Definitions.....37

    Adjustment Code F/M Added Field Definitions.....37

    Ship Via File Maintenance Added Field Definitions.....38

    Service Code File Maintenance Added Field Definitions.....39

    Cash Receipts and Adjustment Entry Payment Screen.....40

    Cash Receipts and Adjustment Entry Tax Adjustment Screen.....41

    Sales Tax Register Screen.....42

    Quote Processing.....42

    Order Processing.....43

    Day End Processing.....44

APPENDIX A - Avalara Support.....45

APPENDIX B – Error Messages.....46

APPENDIX C – Generic Tax Calculation.....47

APPENDIX D – Features by FACTS Version.....48

APPENDIX E – Partner Modification Hook Points.....49

APPENDIX F - DOU100 Bug Fix.....57

APPENDIX G – Software Port Assignments.....58

## Table of Figures

Figure 1 - Avalara Customer Center Log In Screen.....	7
Figure 2 – Sales Tax System Menu.....	9
Figure 3 – AvaTax Control F/M – Settings Tab.....	10
Figure 4 – AvaTax Control F/M – Options Tab.....	11
Figure 5 – AvaTax Control F/M – ECM Tab.....	13
Figure 6 – AvaTax Control F/M – Communications Tab.....	15
Figure 7 – AvaTax Control F/M – Basis Tab.....	16
Figure 8 – AvaTax Control F/M – Statistics Tab.....	16
Figure 9 – Options by State F/M.....	17
Figure 10 - Branch Address F/M.....	18
Figure 11 – AvaTax Tax Code F/M.....	19
Figure 12 – Obsolete Tax Codes.....	20
Figure 13 – AvaTax Entity Use Code F/M.....	20
Figure 14 – Customer/Item Configuration.....	21
Figure 15 – Exemption Certificate Update.....	23
Figure 16 – Calculate Taxes.....	24
Figure 17 – Commit Invoices to AvaTax.....	25
Figure 18 – AvaTax History Removal.....	26
Figure 19 – Verified Address F/M.....	26
Figure 20 – Verify Addresses.....	27
Figure 21 – Address Verification Log.....	28
Figure 22 – Post Past Invoices.....	29

Figure 23 – Tax Detail Inquiry.....	30
Figure 24 – Sales Tax Adjustment Register.....	31
Figure 25 – Sales Order Static Control F/M.....	33
Figure 26 – AR Invoice Processing Control F/M.....	34
Figure 27 – Customer F/M.....	35
Figure 28 – City Correction F/M.....	35
Figure 29 – Item F/M.....	36
Figure 30 – Item Class F/M.....	37
Figure 31 – Adjustment Code F/M.....	37
Figure 32 – Ship Via F/M.....	38
Figure 33 – Service Code F/M.....	39
Figure 34 – Cash Receipts and Adjustment Entry Payment.....	40
Figure 35 – Cash Receipts and Adjustment Entry Tax Adjustment.....	41
Figure 36 – Sales Tax Register.....	42
Figure 37 – Sales Order Entry Line Detail for Use Tax.....	43

## INTRODUCTION

Congratulations! You've just purchased the most complete sales tax solution available for FACTS. Using the AvaTax service, the FACTS Connector provides sales tax calculations for all 50 U.S. States, District of Columbia, and all Canadian Provinces and Territories (including GST, PST, and HST). This activation and configuration guide is broken into sections and provides direction for the following elements:

- Activating Your Avalara Account: An Overview
- The Avalara Connector and Your ERP: An Overview
- Connector Installation
- Connector Integration
- Connector Processing within the ERP
- Connector Administration: A Detailed Guide
- The Connector and Your ERP: A Detailed Guide

## ACTIVATING YOUR AVALARA ACCOUNT: AN OVERVIEW

Upon completion of your organization's order with Avalara, an email from [websales@avalara.com](mailto:websales@avalara.com) is forwarded with confirmation of the order for service. The email provides access information for the Avalara Customer Center, including the following information:

- Avalara Customer Center URL – <https://www.avalara.com/Login>
- Account number
- License key
- Username
- Temporary Password

You will have an Avalara Go Live Consultant to assist you with this activity. Have all this information handy as it will be required entry in the connector setup to access Avalara servers.

**Note:** the connector software itself has no activation key.

### Account Activation

To activate your organization's Avalara Account, access the Avalara User Center (<https://www.avalara.com/Login>), and log in using the username and password provided in the above referenced email from [websales@avalara.com](mailto:websales@avalara.com).

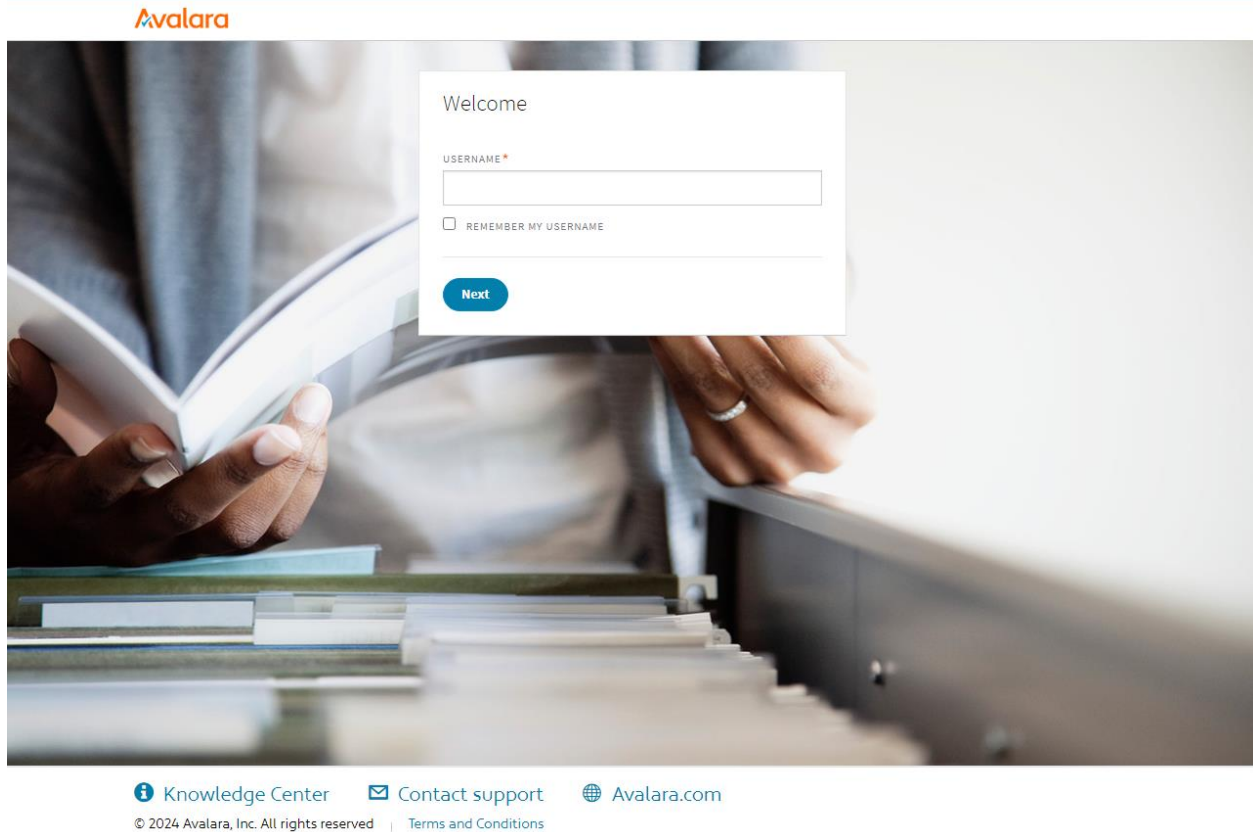


Figure 1 The Avalara Customer Center Log In Screen.

## THE AVALARA CONNECTOR AND YOUR ERP: AN OVERVIEW

This section provides you with a high-level overview of what to expect once you've purchased the **FACTS AvaTax Connector** and is divided into two parts:

- The FACTS AvaTax Connector Installation Process
- How the FACTS AvaTax Connector Works within your ERP

### The FACTS AvaTax Connector Installation Process

With assistance from Sequoia Group, Inc., your solution partner will install the FACTS AvaTax Connector. We recommend that it first be installed in a test environment for full review. Testing will be performed in the FACTS test environment and when complete, the project manager will coordinate installation in the Production environment. You will be able to run transactions through your Live FACTS environment and have the AvaTax service in test mode. Once testing is completed and is signed off, only then will the AvaTax service be put into production.



## How the AvaTax Connector Works within FACTS

The connection between FACTS and AvaTax is transparent and comprehensive. Every program across the AR, SO and SR modules that require tax calculation, address validation or certificate management triggers the connector to send or receive the necessary information to and from AvaTax. Avalara's fees are based upon transaction volume: the greater of number of either invoices or API calls divided by ten. Every ten API calls for address validation *outside* tax calculation also count as a transaction. Review Section 3 on their website for definition updates:

<https://www.avalara.com/us/en/legal/terms.html>

The date of the transaction is evaluated for each calculation as changes to rates may occur over time. Sales orders use entry date while confirmation uses invoice date. Credit memos use either invoice or credit memo date depending upon a system setting. Starting with release 7.8, orders containing line items with negative quantities, i.e. items returned during a sale, reference "Usage Date" to determine taxability of when that item was originally invoiced. This ensures that the calculated tax mirrors the originally billed amount.

Use Tax as recognized through the sales order process for items consumed during a project are also reported. Address verification also works with other modules like Accounts Payable, Payroll, Office Automation and Relationship Management.

The Tax Table file maintenance is still used by the connector, but solely for general ledger purposes. Customers and ship-tos must be assigned to a valid tax code. All three segments: state, county, and local are supported. The other fields are ignored. Tax jurisdiction and rates are determined by Avalara using precise origin and destination addresses.

While the system of record reporting moves to the Avalara Web Console for its complete and more detailed presentation, standard FACTS tax reporting functions as it did previously but only to a limited number of tax categories. State and County taxes match the actual calculated values. If there are transit, sports or other taxes below "Local", however, they are lumped in with the local tax. It provides the ability to separate recognition of committed vs uncommitted transactions for those who use Cash Basis tax reporting.

Some features found in this manual were added subsequent to the launch at version 7.7. Support for versions back to 7.5 was retrofitted with the base feature set. Please refer to Appendix D at the end of the document for a complete list

## CONNECTOR ADMINISTRATION

Once your **FACTS AvaTax Connector** has been installed, you may access the Sales Tax System using access code "TX" from any menu.

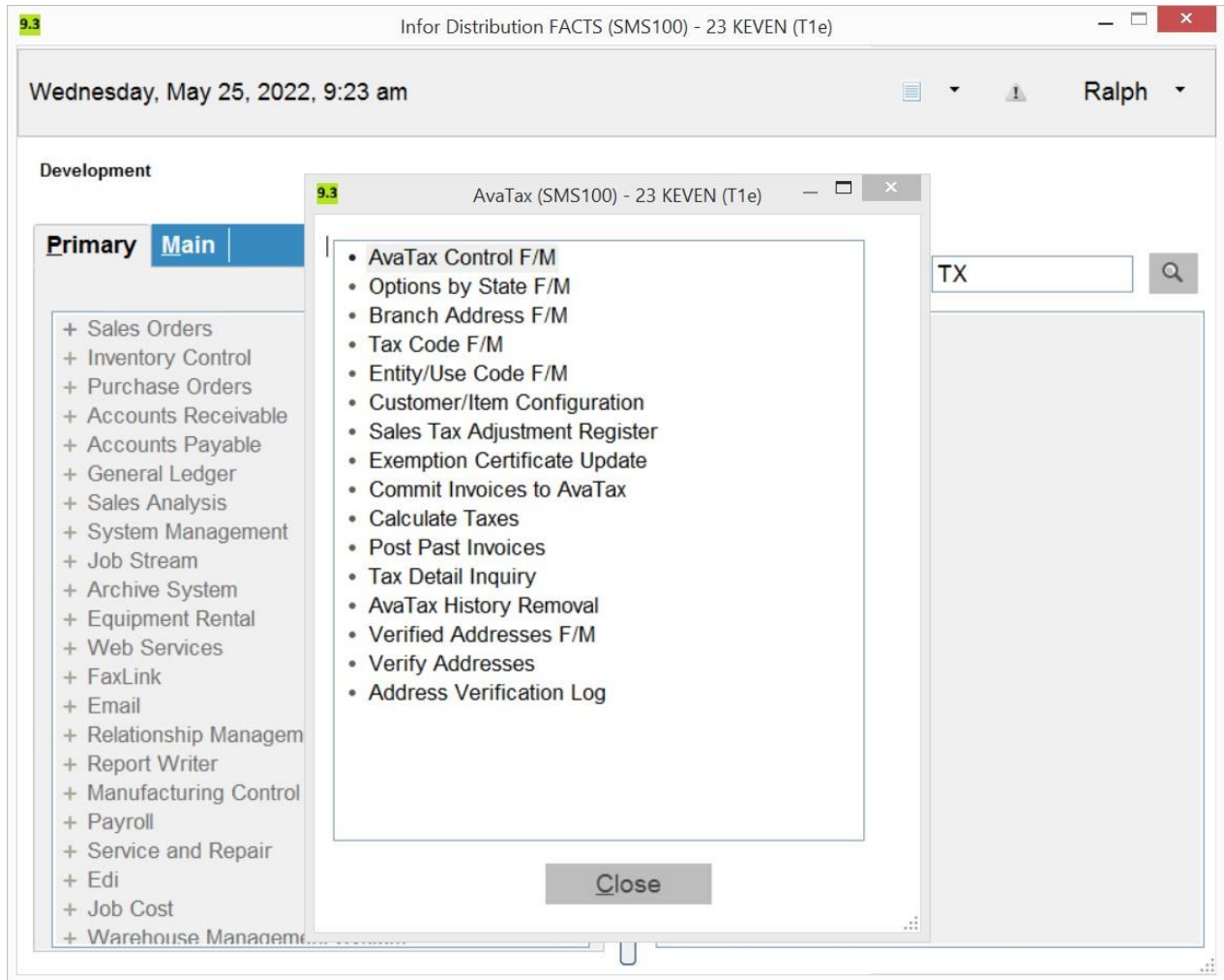


Figure 2. The Sales Tax System menu is accessible using access code "TX".

## AvaTax Control F/M (TXF990)

Figure 3. AvaTax Control F/M - Settings tab.

Here you find primary settings for the connector.

### AvaTax Control F/M - Settings Tab Field Definitions

Field Name	Definition	Valid Values
<b>AvaTax Company</b>	Identifies the Company Code defined in the Avalara Administration portal. We recommend using the same company code used by your FACTS system.	Valid up to 20 character alpha numeric value. If left blank, it will use the FACTS company number.
<b>Time Out</b>	The number of seconds to wait from a response from Avalara's servers.	A non-zero numeric value. We recommend starting with 15.
<b>Years of History</b>	Enter the number of years to retain detailed history of Avalara response codes for each transaction. Useful for audits.	A numeric value from 0 to 99.
<b>CM Tax Calc Date</b>	Determines which date is used for tax calculations on a Credit Memo. The other date is used as reporting period.	I-Use Referenced Invoice Date M-Use This Credit Memo Date
<b>Default Freight Code</b>	Determines default code used unless specific value configured in Ship Via F/M	Blank= FR020100 "Shipping only common carrier - FOB destination"

<b>Default Freight/ Handling Code</b>	Determines default code used when a handling fee is present in Ship Via F/M unless a specific value is configured	Blank= FR030000 "Shipping And Handling Combined"
---	---	--

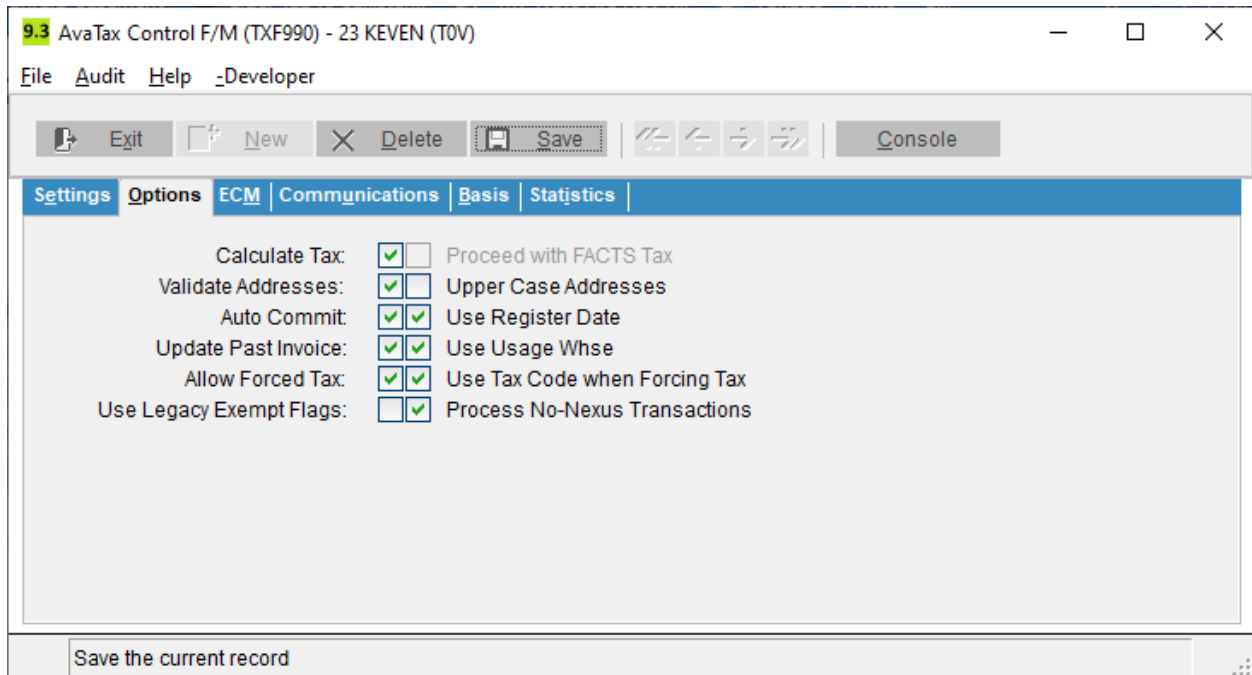


Figure 4. AvaTax Control F/M - Options tab.

This tab determines options that control the connector's functionality.

### AvaTax Control F/M - Options Tab Field Definitions

Field Name	Definition	Valid Values
<b>Calculate Tax</b>	Allows communications with AvaTax to be turned off if network is down. See "Calculate Taxes" program description on page 24.	Default=checked Offline mode=unchecked
<b>Proceed with FACTS Tax</b>	Allow invoices to be printed with tax calculated by FACTS when AvaTax is turned off. Useful for counter sales. Invoices must be updated by "Calculate Taxes" before they can be processed by the DSR. Note: Printed invoice total and AR total may not match!	Default=Unchecked

<b>Validate Addresses</b>	Enables use of the Avalara address verification service. Uncheck if using the Sequoia Group hosted service.	Default=checked
<b>Use Legacy Exempt Flags</b>	If checked, FACTS will honor the customer exempt flag (X) on invoices. It will also honor the item taxable flag <i>unless an</i> Avalara Tax Code is specified for the item or item's class.	Default=Unchecked
<b>Auto Commit</b>	Enabling auto update runs the "Commit Invoices to AvaTax" program automatically following the running of various registers.	Default=Unchecked
<b>Upper Case Addresses</b>	Checked-Force address verification to use all upper case.	Default=Checked Mixed case=unchecked
<b>Update Past Invoice</b>	Update sales tax information on past invoice document when tax is adjusted in cash receipts.	Default=Unchecked
<b>Allow Forced Tax</b>	This allows you to force tax on an invoice and suppresses the item Tax Code. Also set tax rate in document footer to "L"	Default=Unchecked
<b>Use Usage Whse</b>	Use usage warehouse as origin address for credits. This helps ensure the tax credit is the same as the original invoice, especially if the invoice was will-call.	Default=Unchecked
<b>Use Register Date</b>	Uses register date when invoice date falls outside month and year of register date. Works with sales registers to ensure posting to the current reporting period in AvaTax.	Default=Unchecked
<b>Use Tax Code when Forcing Tax</b>	This is used in conjunction with the "Forced Tax" option. In this case, it sends the item Tax Code to ensure proper calculation in some instances.	Default=Unchecked

<b>Process No-Nexus Transactions</b>	Normally, the connector will not send a document to AvaTax if it sees that you are not registered to collect sales tax in the transaction state. If you turn this option on all documents will be sent to AvaTax for processing.	Default=Unchecked
--------------------------------------	--	-------------------

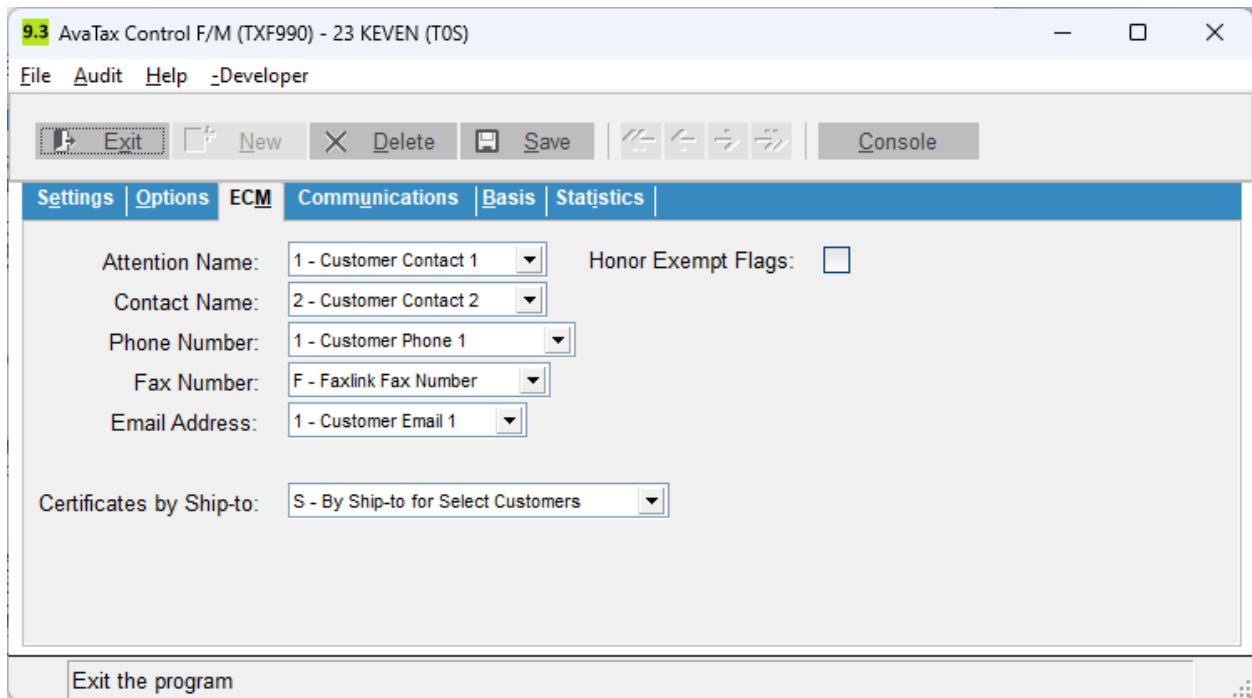


Figure 5. AvaTax Control F/M - ECM tab.

This tab sets default values for the uploading of customer information to Avalara for use with Exemption Certificate Management as well as some ECM parameters.

### AvaTax Control F/M - ECM Tab Field Definitions

Field Name	Definition	Valid Values
<b>Attention Name</b>	“Attention” contact name for setting Exemption Certificate default.	Contact 1, Contact 2, Faxlink Contact, None
<b>Contact Name</b>	Primary contact name for setting Exemption Certificate default.	Contact 1, Contact 2, Faxlink Contact, None
<b>Phone Number</b>	Customer primary phone for setting Exemption Certificate default.	Phone 1, Phone 2, Faxlink #, None

<b>Fax Number</b>	Customer fax number for setting Exemption Certificate default.	Phone 1, Phone 2, Faxlink #, None
<b>Email Address</b>	Customer email address for setting Exemption Certificate default.	Phone 1, Phone 2, Faxlink #, None
<b>Honor Exempt Flags</b>	If checked, only active customers and permanent ship-tos with Tax Rate flag set to "X" will be uploaded to Avalara.	Default=Unchecked
<b>Certificates by Ship-to</b>	<p>Options are:</p> <p>N-By Customer Account which means customers will be uploaded to ECM with exposure states determined by customer's ship-tos.</p> <p>Y-By Ship-to for All Customers which means that the customer account will be uploaded to ECM and a separate customer account will be uploaded to ECM for each ship-to. This allows for exempt certificates for a specific ship-to. The ship-to customer number in ECM will be CustomerNumber~~ShipTo.</p> <p>S-By Ship-to for Select Customers which is the same as option Y above except that a customer account will only be split out by ship-to in ECM if the customer is set to do so in Customer/Item Configuration in FACTS.</p>	Default=N-By Customer Account

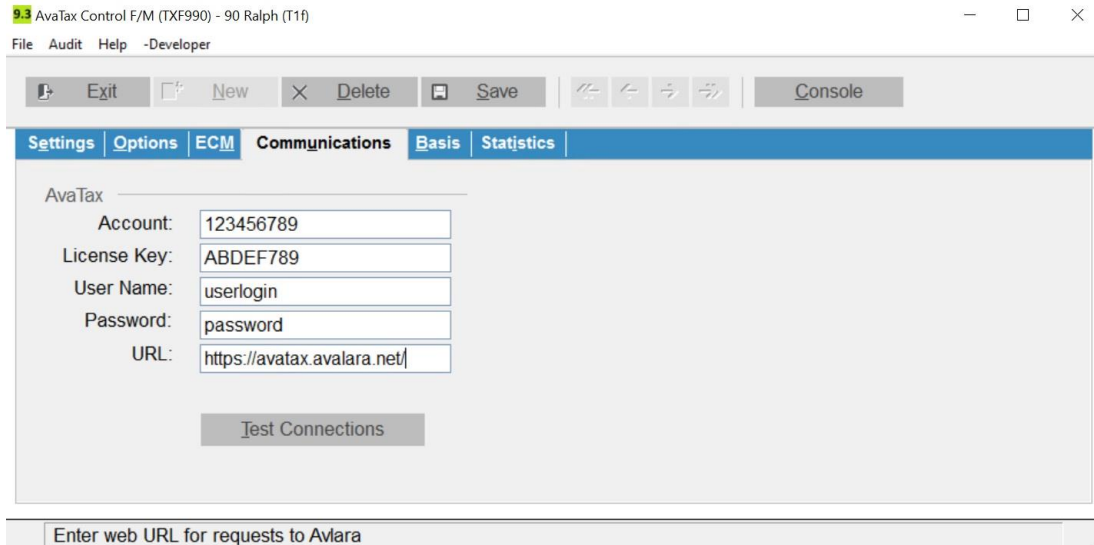


Figure 6. AvaTax Control F/M - Communications tab.

This tab gives access to *required* login credentials provided by Avalara, provides a means to verify connectivity to Avalara Services and allows you to link directly to the Avalara Administration Console. Note the following URLs:

AvaTax Live: <https://avatax.avalara.net>

AvaTax Sandbox: <https://development.avalara.net>

### AvaTax Control F/M - Communications Tab Field Definitions

Button	Definition
<b>Account</b>	Account number assigned by Avalara
<b>License Key</b>	Provided by Avalara, but new one can be created on AvaTax Console
<b>AvaTax User Name</b>	Your Avalara user name
<b>AvaTax Password</b>	Your Avalara login password
<b>AvaTax URL</b>	Web address for either Live or Sandbox operation
<b>Test Connection button</b>	Accesses Avalara website and verifies your login credentials. You should get messages below the button with status and if successful, the current version number and TLS revision.
<b>Console button</b>	Launches your workstation web browser and takes you to the Avalara Console login screen.



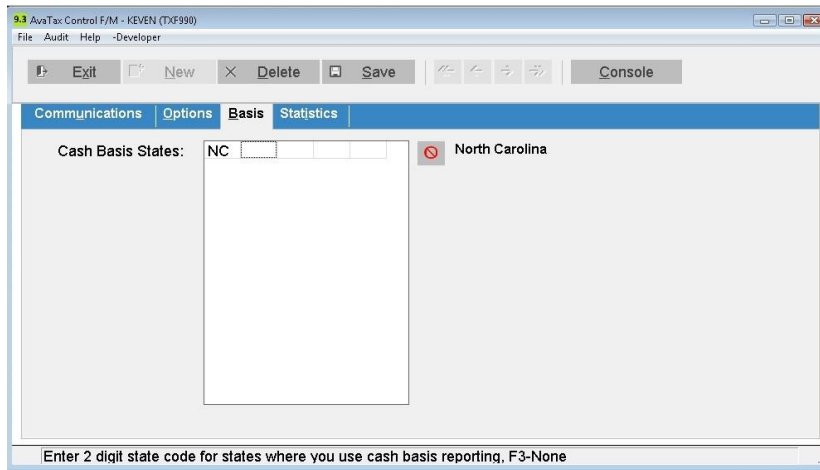


Figure 7. AvaTax Control F/M - Basis Tab

This tab allows you to designate in which state(s) you report tax on a cash basis. The FACTS sales tax register has been enhanced to provide auditing capabilities to support this feature.

### AvaTax Control F/M - Basis Tab Field Definitions

Field Name	Definition	Valid Values
<b>States</b>	Enter a list of states for which you report tax on a cash basis. <b>Note: Introduced at 7.9.</b>	Valid two character state codes

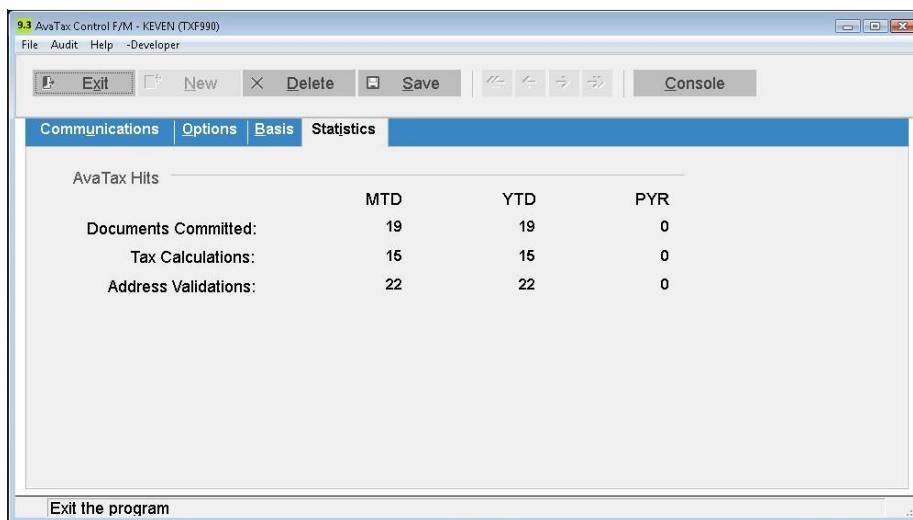


Figure 8. AvaTax Control F/M - Statistics Tab

This tab allows you to view the number of occurrences or “hits” for various statistics: Documents Committed, Tax Calculations and Address Validations for Month to Date, Year to Date and Previous Year to Date periods of time. Tax calculation API calls are made only for transactions within Nexus states as configured on the AvaTax console.

These statistics are meant to represent our best estimate of those used by Avalara to determine billing volumes. We continually work to optimize the connector for the most efficient use of resources. **Note: Introduced at 9.3.**

 Quantity of One:'. A status bar at the bottom reads 'Send quantity of one and full extended price to AvaTax'."/>

Figure 9. Options by State F/M

This maintenance controls state specific options. **Note: Introduced at 9.3**

### Options by State F/M - Field Definitions

Field Name	Definition	Valid Values
<b>Country Code</b>	Identifies the country	Two character country code
<b>State Code</b>	Identifies the state or province	Two character state code.
<b>Adjust Tax for Early Pay Discounts</b>	Does state or province tax adjustments for early pay discounts?	Default = unchecked
<b>Quantity of One</b>	Sends quantity of one with full extension to accommodate Florida Discretionary Surtax	Default = unchecked

Figure 10. The Branch Address F/M identifies the origin address for transactions that do not involve a warehouse (AR Invoicing) and enables the ability to post tax by branch as required by some states.

### Branch Address F/M Field Definitions

Field Name	Definition	Valid Values
<b>Address 1</b>	First line of address.	
<b>Address 2</b>	Second line of address if needed.	
<b>City</b>	City	
<b>State</b>	State	Valid two digit state code
<b>Zip Code</b>	Zip code	
<b>Address 3</b>	Expanded full address	Click icon to build automatically
<b>Location Code</b>	Avalara provided value used for providing branch specific reporting required by some states. Determined by initiating warehouse	As defined on the AvaTax Console. Entry is optional
<b>AvaTax Company</b>	Avalara company code (if different) <b>Note: Introduced at 9.1</b>	Entry optional

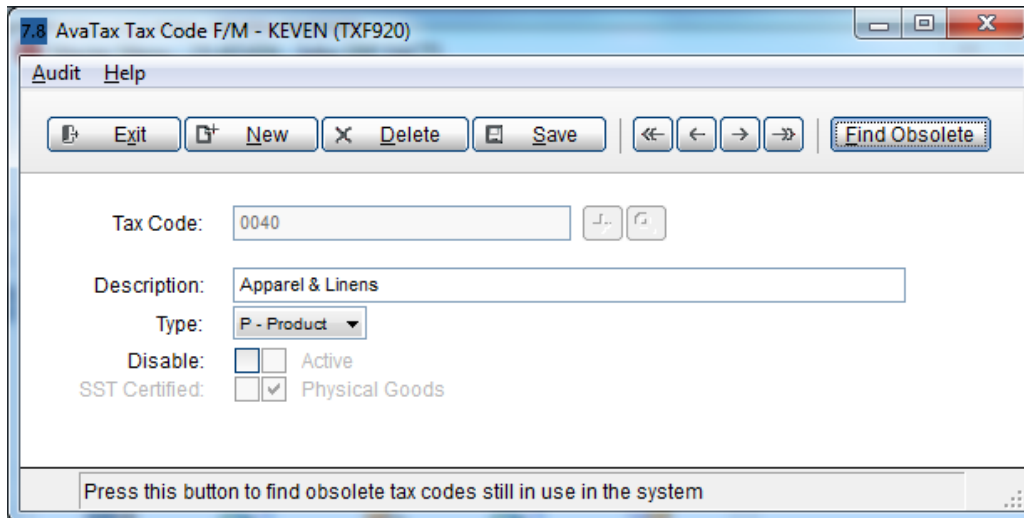


Figure 11. AvaTax Tax Code F/M

This is a table of Avalara supplied codes used to categorize the goods and services found in your item file. You will need to initially download them at time of installation using the “F1-Download codes” option.

### AvaTax Tax Code F/M Field Definitions

Field Name	Definition	Valid Values
<b>Tax Code</b>	Avalara tax code	F1 - Download codes directly from Avalara website (this must be performed initially)
<b>Description</b>	Tax code description	Supplied by Avalara
<b>Type</b>	Avalara tax code type	P-Product F-Freight D-Digital S-Service O-Other
<b>Disable</b>	If disabled, the code will show up only in the file maintenance.	Default=Unchecked
<b>Active</b>	Active tax codes	Default=unchecked
<b>SST Certified</b>	Streamlined Sales Tax Certified	Default=unchecked
<b>Physical Goods</b>	Tax codes related to physical goods	Default=checked

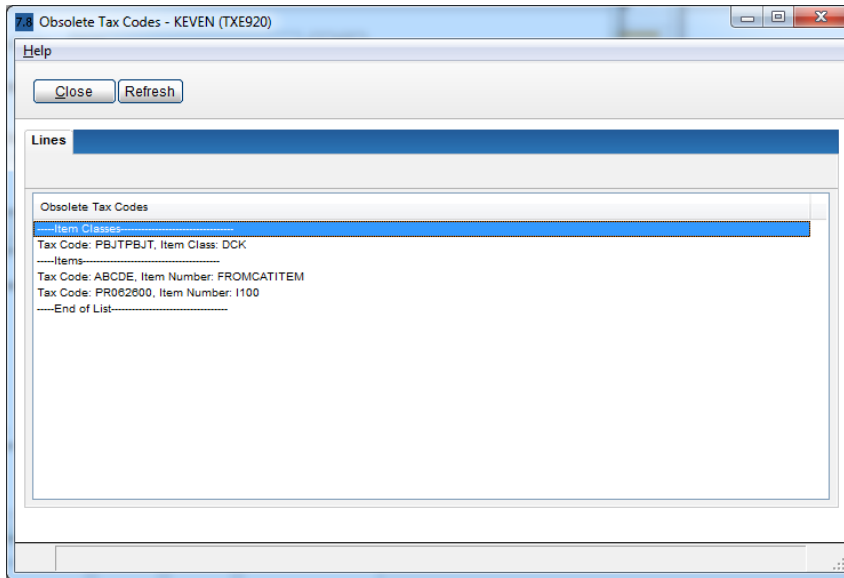


Figure 12. Obsolete Tax Codes

The “Find Obsolete” button will locate and display and items, classes, etc. that have tax codes which are either no longer on file, disabled, or inactive. The view includes a refresh button so that the user may monitor progress as they correct those codes. **Note: introduced at 7.8**

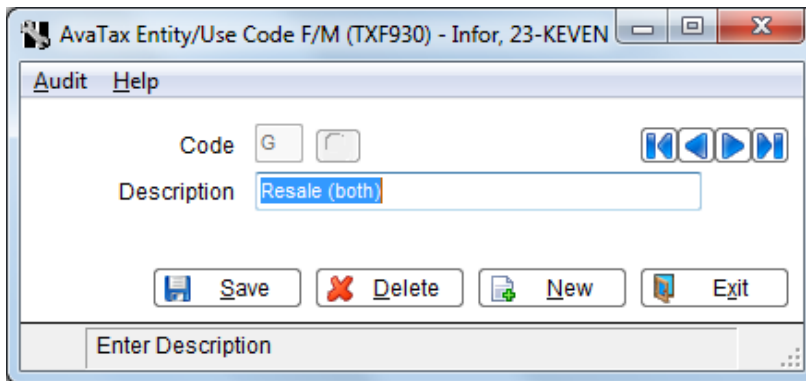


Figure 13. AvaTax Entity/Use Code F/M

This is a table of Avalara supplied codes which are used to categorize your customers and individual ship-to locations. We deliver the connector with the file pre-populated.

### AvaTax Entity/Use Code F/M Field Definitions

Field Name	Definition	Valid Values
<b>Code</b>	Avalara code	All codes are preloaded
<b>Description</b>	Entity/Use code description	Supplied by Avalara

9.3 Customer/Item Configuration (TXE910) - 23 KEVEN (TOS)

File Options Help

Done Import Lines

Customer Class:  All Classes

Customer:  C100 SOUTHEASTERN INDUSTRIAL SUPPLY

Ship-To:  All Ship-tos

Tax Consumables:

ECM by Ship-to:

Entity/Use Code:  No exempt code

Effective:  Now

Latitude:  40.6698200

Longitude:  -111.8787600

AvaTax Customer:

Lines

+

Item Class:

Item:

Entity/Use Code:

Item	Class	Code	Description
add			

Figure 14. Customer/Item Configuration

This configuration controls the entity/use code at the customer level. It can be set at a customer class level or may be specific to a customer, a customer / ship-to combination. A customer may fit into one entity code for most items while having a different code for a particular item or group of items such as tools or consumables. They can either be non-taxable or taxable exceptions. And for some situations like Georgia where freight is taxable when any line is taxable, you may enter "All Item Classes" as an exception. It may also include latitude and longitude coordinates for rural areas and construction sites where address verification is unavailable and overridden. The "Import" button allows for setting up a new customer, class, or cust/ship-to by using an existing record's configuration. The lines are imported and will replace any matching items. **Note: Import button functionality introduced at 7.9 and Lat/Long geolocation introduced at 9.1.**

**Important consideration:** When entity codes are used for the purpose of exemption for a series of item(s) or an item class for that customer scope, it is important that under the "Settings" tab of the AvaTax console, you select the "Optional" radio button for the United States and for Canada where that applies. Otherwise setting the Exemption Certificate option to "Required" will cause AvaTax to ignore most entity codes.

**Avalara AvaTax** Account#:

Home Organization Transactions Tax Returns Reports Tools **Settings** Avalara Cust

### Exemption Certificate Options

**Save** **Cancel**

Enable Exemption Certificate Management System (ECMS) [?](#)

---

United States:  Ignored  Optional  Required

Canada:  Ignored  Optional  Required

---

Exemption Override Code:  [?](#)

## Customer/Item F/M Field Definitions

Field Name	Definition	Valid Values
<b>Customer Class</b>	FACTS customer class	Any Customer class F1- All classes, F2-Class Search, F3- Record Search
<b>Customer</b>	FACTS customer	Any Customer
<b>Ship-To</b>	FACTS Ship-To	Any Ship-to
<b>Entity/Use Code</b>	Avalara code	Any Avalara Entity/Use code
<b>AvaTax Customer Number</b>	Avalara assigned number	Used if FACTS customer code changes after initial upload.
<b>Effective Date</b>	Beginning date for using this exception record	Any date expression F1 = Today
<b>AvaTax Customer</b>	If the customer is already entered into the AvaTax system but with a customer number different from the FACTS customer number, you can enter the AvaTax customer number here to allow for a connection between the account in FACTS and	Any Valid AvaTax Customer Number  Note: this is rarely used

	the account in AvaTax.	
<b>Tax Consumables</b>	Determines whether or not items flagged as consumables are always taxed	Default = unchecked
<b>ECM by Ship-to</b>	If the ECM by Ship-to option in AvaTax Control F/M is set to S-By Ship-to for Select Customers this is how to set that this customer should be uploaded with a separate account for each ship-to.	Default = unchecked
<b>Latitude</b>	Degrees of latitude	Optional entry overrides address
<b>Longitude</b>	Degrees of longitude	Optional entry overrides address
<b>Item Class</b>	FACTS Item Class	Any item class
<b>Item</b>	FACTS Item	Any item
<b>Entity/Use Code</b>	Avalara code	Any Avalara Entity/Use code

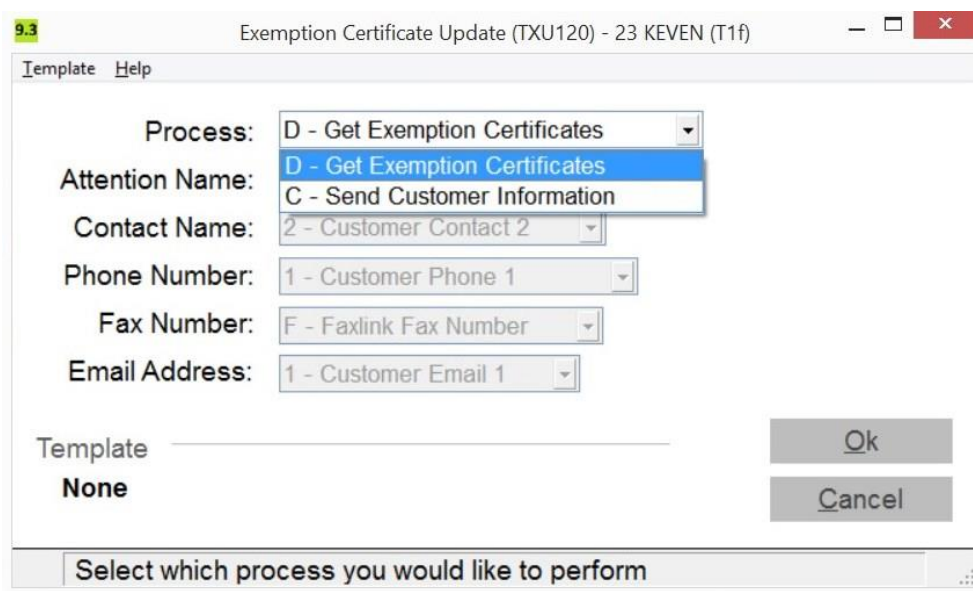


Figure 15. Update for uploading customer data to ECM or downloading exemption certificate information into FACTS.



Note: This is normally used only to upload customer accounts when first setting up AvaTax in FACTS. After initial upload, customer additions or changes in FACTS will automatically be uploaded to ECM. Also, when viewing exemption certificate information in Customer Inquiry, ECM will be contacted 'real-time' so you are sure to be viewing the most current data.

### Exemption Certificate Update Field Definitions

Field Name	Definition	Valid Values
<b>Process</b>	Determines whether to upload customer data or download exemption results.	C-Send Customer Information D-Get Exemption Certificates
<b>Attention Name</b>	"Attention" contact name	Contact 1, Contact 2, Faxlink Contact, None
<b>Contact Name</b>	Primary contact name	Contact 1, Contact 2, Faxlink Contact, None
<b>Phone Number</b>	Customer primary phone	Phone 1, Phone 2, Faxlink #, None
<b>Fax Number</b>	Customer fax number	Phone 1, Phone 2, Faxlink #, None
<b>Email Address</b>	Customer email address	Phone 1, Phone 2, Faxlink #, None

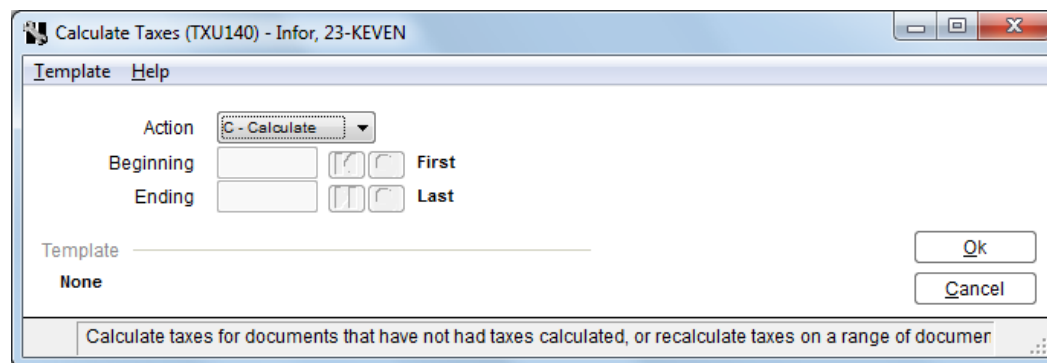


Figure 16. Calculate Taxes updates documents entered while system is offline to Avalara.

The FACTS AvaTax Connector allows for offline processing if there is network difficulty communicating with the AvaTax service. Tax calculation may be temporarily disabled for order processing via the "Calculate Taxes" flag found on the options tab of AvaTax Control F/M. Order entry may continue although without tax calculation. When the service is available, the

calculation may be re-enabled. This update will calculate taxes for documents that have not had taxes calculated.

### Calculate Taxes Option Definitions

Field Name	Definition	Valid Values
<b>Action</b>	Calculates taxes on new documents or recalculates a range of documents.	C-Calculate R-Recalculate
<b>Beginning</b>	Beginning document number for recalculation option only.	Six digit document number
<b>Ending</b>	Ending document number for recalculation option only.	Six digit document number

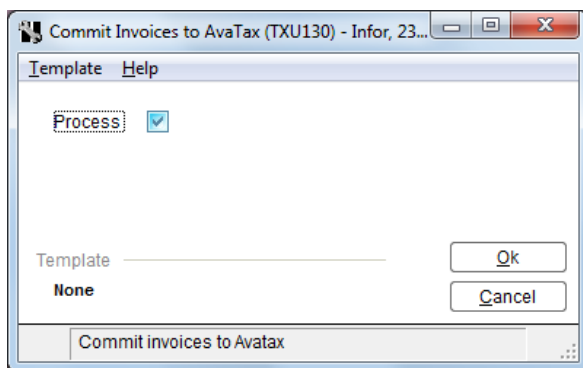


Figure 17. Commit Invoices to AvaTax.

The final phase of transaction handling is to commit invoices to Avalara. Any subsequent corrections or adjustments must be processed using Cash Receipts adjustment entry. Also, it can be run automatically following the updating of sales registers in both AR and SO along with Cash Receipts & Adjustments if the "Auto Update" flag is set in AvaTax Control F/M on the options tab. Click Ok button to run. Commit date is invoice date or system date for cash basis transactions.

### Commit Invoices to AvaTax Option Definitions

Field Name	Definition	Valid Values
<b>Process</b>	Upload commit flag to Avalara	Y or N

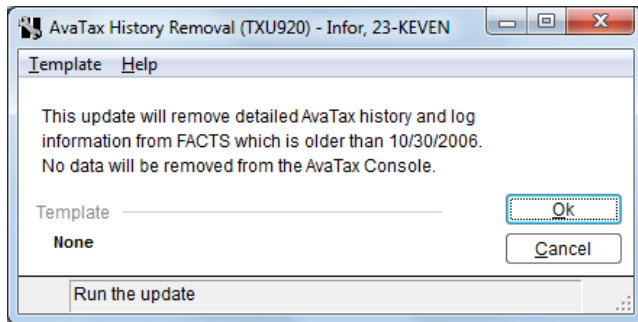


Figure 18. AvaTax History Removal

This update removes detailed transaction history and log information maintained within the FACTS database older than the "Years of History" specified in the AvaTax Control F/M on the settings tab. Click Ok button to run or cancel to exit.

### AvaTax History Removal Definitions

Button	Definition
Ok	Remove transactions dated prior to "Years of History" cutoff

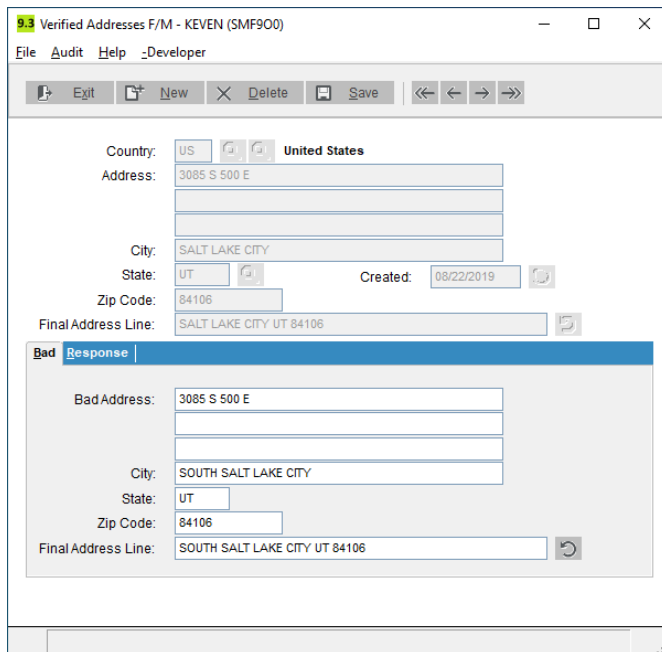


Figure 19. Verified Addresses F/M

This program/file serves a 'cache' for address verification. The system will automatically cache both the original and verified address and retain the date of correction. These entries will be retained for 180 days to minimize hits to the AvaTax address servers. Older records will be ignored, so that any changes, etc. will be correctly honored.

This program/file can also be used to 'force' an address. We have seen a few incidences where some valid addresses are not verified. If that occurs, the user may enter the address in the "Bad" tab so that it will use the address from this file and not hit the address servers. Records entered manually never expire. The "Response" tab shows the XML Avalara returned.

An alternative way to force an "unverified" address is to fill in the City, State, Zip and include the text "General Delivery" either at the end of the Address 1 line or in Address 2. The connector will return the tax for that zip code as opposed to the specific street address where there may be none.

### Verified Addresses F/M Definitions

Field Name	Definition	Valid Values
<b>Country</b>	Enter country code	Valid country code
<b>Address</b>	Enter address information	Any text
<b>City</b>	Enter city	Any text
<b>State</b>	Enter state	Valid state code
<b>Zip Code</b>	Enter zip code	Numeric
<b>Final Address Line</b>	Enter city state zip	Any text
<b>Created</b>	Date system captured address	Valid date

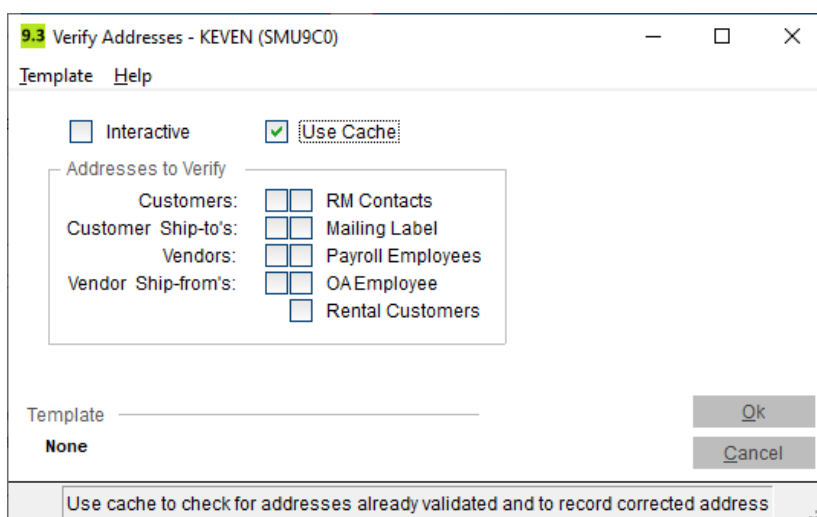


Figure 20. Verify Addresses

This update is used to initially verify and correct existing addresses in all FACTS files during implementation. Every subsequent addition or edit automatically launches the verification process.

### Verify Addresses Option Definitions

Field Name	Definition	Valid Values
<b>Interactive</b>	Ability to monitor exceptions during update. Alternately, you may run the Address Verification Log Report to locate any exceptions	Y or N
<b>Addresses to Verify</b>	Selection of files available to update. Choose each one to be verified during process.	Y or N
<b>Use Cache</b>	Selecting this option updates only addresses not found in current address verification file. Can greatly minimize transactional traffic.	Y or N

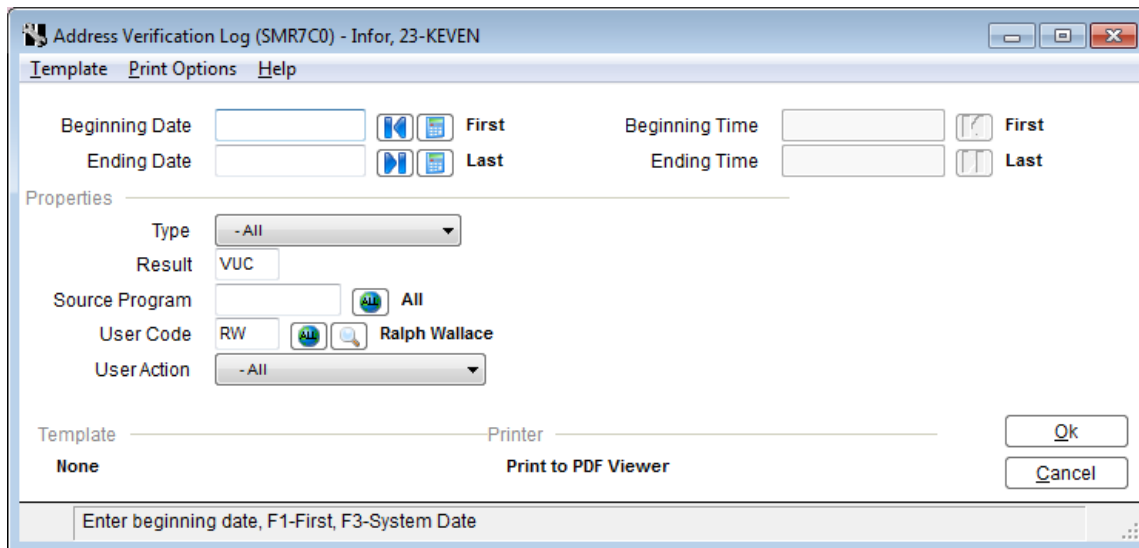


Figure 21. Address Verification Log

This log report details all address verification changes made to the various FACTS files. Many filtering options are available. Note the following result codes:

- 0 Address already correct
- 25 Address verified and corrected
- 50 Error. Address could not be verified and remains unchanged

### Address Verification Log Option Definitions

Field Name	Definition	Valid Values
<b>Beginning Date</b>	Beginning date for inclusion on report	Any date expression
<b>Ending Date</b>	Ending date for inclusion on report	Any date expression
<b>Beginning Time</b>	Beginning time for (only enabled when beginning and ending date are the same)	Time in HH:MM:SS format (optionally add "A" or "P")
<b>Ending Time</b>	Ending time for inclusion on report (only enabled when beginning and ending date are the same)	Time in HH:MM:SS format (optionally add "A" or "P")
<b>Type</b>	Address Type	Select from drop down list or CR=All
<b>Result</b>	Address Result	Enter one or more result codes: V-Verified U-Unverified C-Changed
<b>Source Program</b>	Program name from which address was verified	FACTS program name F1-All
<b>User Code</b>	User Code to show for log entries	Valid user code F1-All
<b>User Action</b>	Action taken by user	All-All changes YES-Accepted changes NO-Rejected changes F4-Rejected via F4

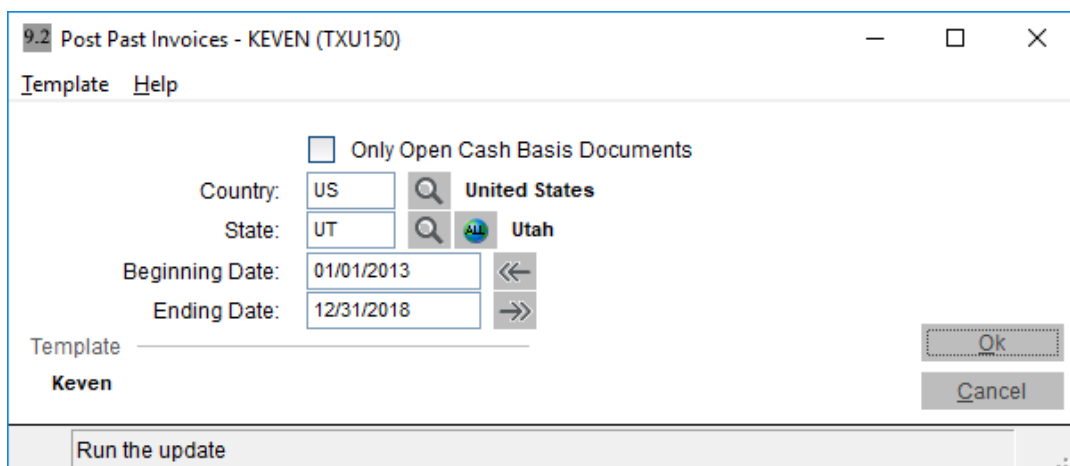


Figure 22. Post Past Invoices

This update will submit prior invoices to the AvaTax console. In the event of implementing the connector in the middle of a reporting period, you can ensure that all invoices for that period are included for tax reporting purposes. If Only Open Cash Basis Documents is selected, it will post all open documents and all other inputs are disabled. **Note: Cash Basis Feature introduced at 9.2.**

### Post Past Invoices Definitions

Field Name	Definition	Valid Values
<b>Only Cash Basis Docs</b>	Used for cash basis only	Optional
<b>Country</b>	Country	Any valid country code
<b>State</b>	State	Optional
<b>Beginning Date</b>	Beginning date to update	Any date expression
<b>Ending Date</b>	Ending date to update	Any date expression following beginning date

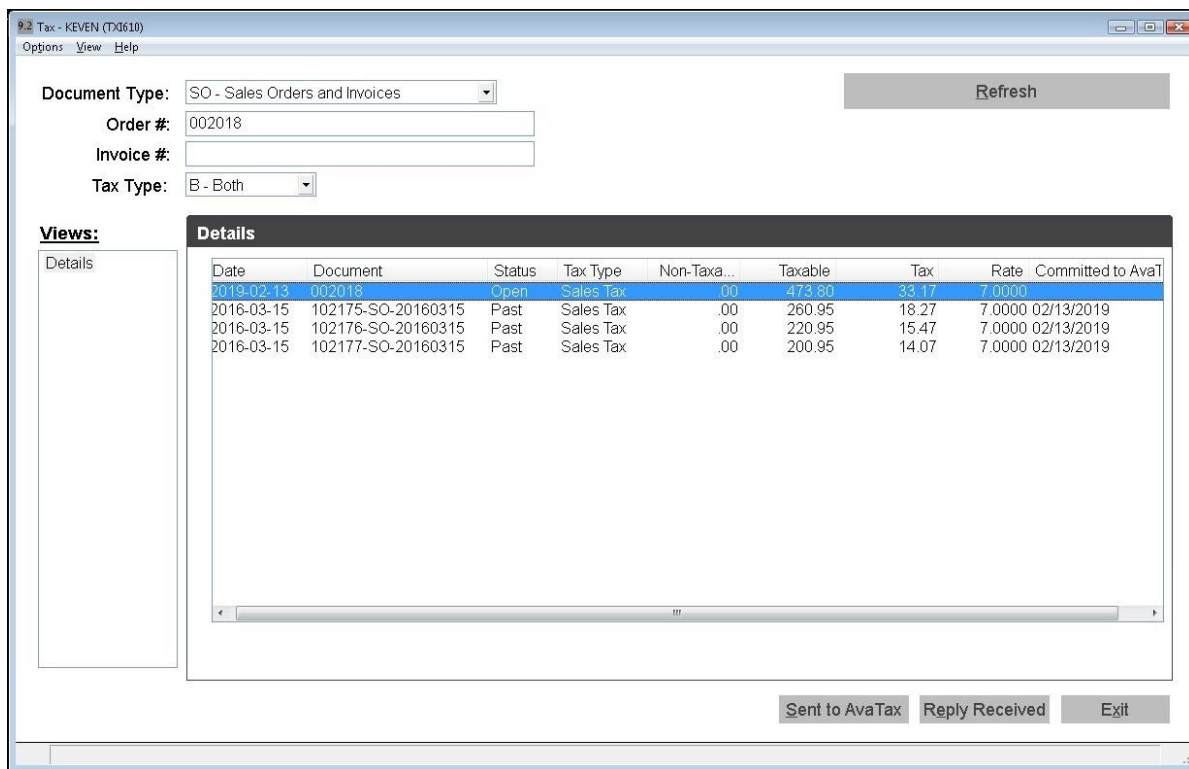


Figure 23. Tax Detail Inquiry

This inquiry provides a detailed view of the transactional dialog between the connector and the Avalara tax service. It begins with a summary display and provides drill downs to what data was sent to Avalara and also what was received from Avalara.

### Detail Tax Inquiry Definitions

Field Name	Definition	Valid Values
<b>Document Type</b>	Document Type selection	SO-Sales Orders/Invoices SQ-Quotes RS-Recurring Sales Orders AI-AR Invoices AR-AR Recurring Invoices SC-Service & Repair Contracts RO-Service & Repair Orders SP-Service & Repair Proposals RR-Equipment Rental Reservation RC-Equipment Rental Contracts
<b>Document Choice (type specific)</b>	Choice of document to display	Any valid transaction number
<b>Document Choice (second tier)</b>	Optional second document	Any valid transaction number
<b>Tax Type</b>	Tax type selection	B-Both (default) S-Sales Tax only U-Use Tax only

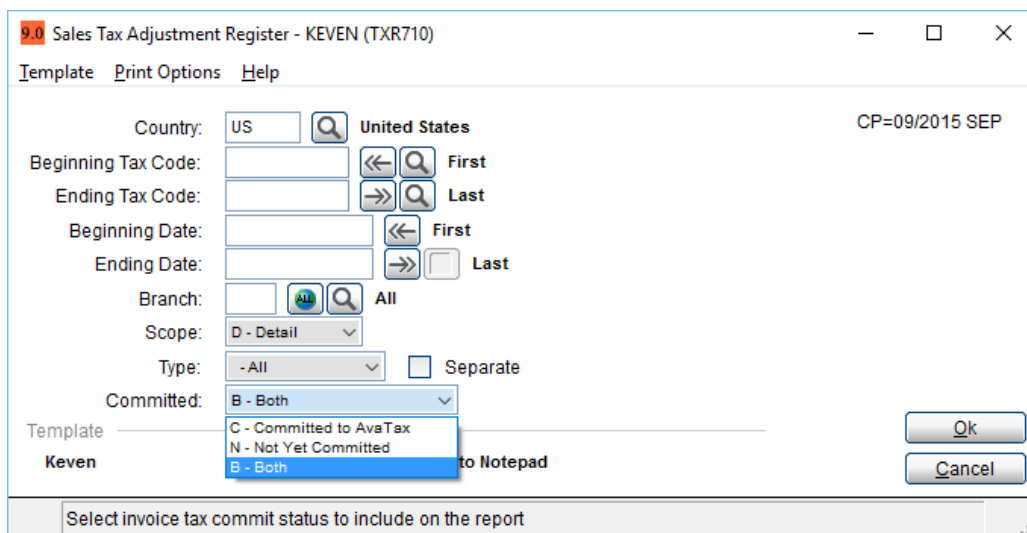


Figure 24. Sales Tax Adjustment Register



This register will print all or selected tax adjustments created by Cash Receipts Entry either for correcting the sales tax amount or by early pay discounts. This report may be used in conjunction with the standard FACTS Sales Tax Register to provide a complete picture of tax collected. One may also select invoices based upon whether or not they have been committed.

**Note: The Committed status is part of the Cash Basis handling introduced at 7.9**

### Sales Tax Adjustment Register Definitions

Field Name	Definition	Valid Values
<b>Country</b>	Country	Any valid country code
<b>Beginning Tax Code</b>	Beginning tax code to print	Default=first
<b>Ending Tax Code</b>	Ending tax code to print	Default=last
<b>Beginning Date</b>	Beginning date to print	Any date expression
<b>Ending Date</b>	Ending date to print	Any date following beginning
<b>Branch</b>	Selected branch to print	Valid code, Default=all
<b>Scope</b>	Detail or Summary	D=Detail, S=Summary
<b>Type</b>	All, Adjustment or Discounts	Default=all
<b>Separate</b>	If checked will print separate reports by type	Default=unchecked
<b>Committed</b>	Determines which invoices are included based upon committed to AvaTax status. <b>Note: Introduced at 7.9.</b>	C-Committed to AvaTax N-Not Yet Committed B-Both

## THE CONNECTOR and YOUR ERP

This section of the documentation details what is different in FACTS now that Avalara is the source for sales tax calculation and reporting. First we'll review key settings found in Sales Order Static Control File Maintenance and Accounts Receivable Invoice Processing File Maintenance that should already be set.

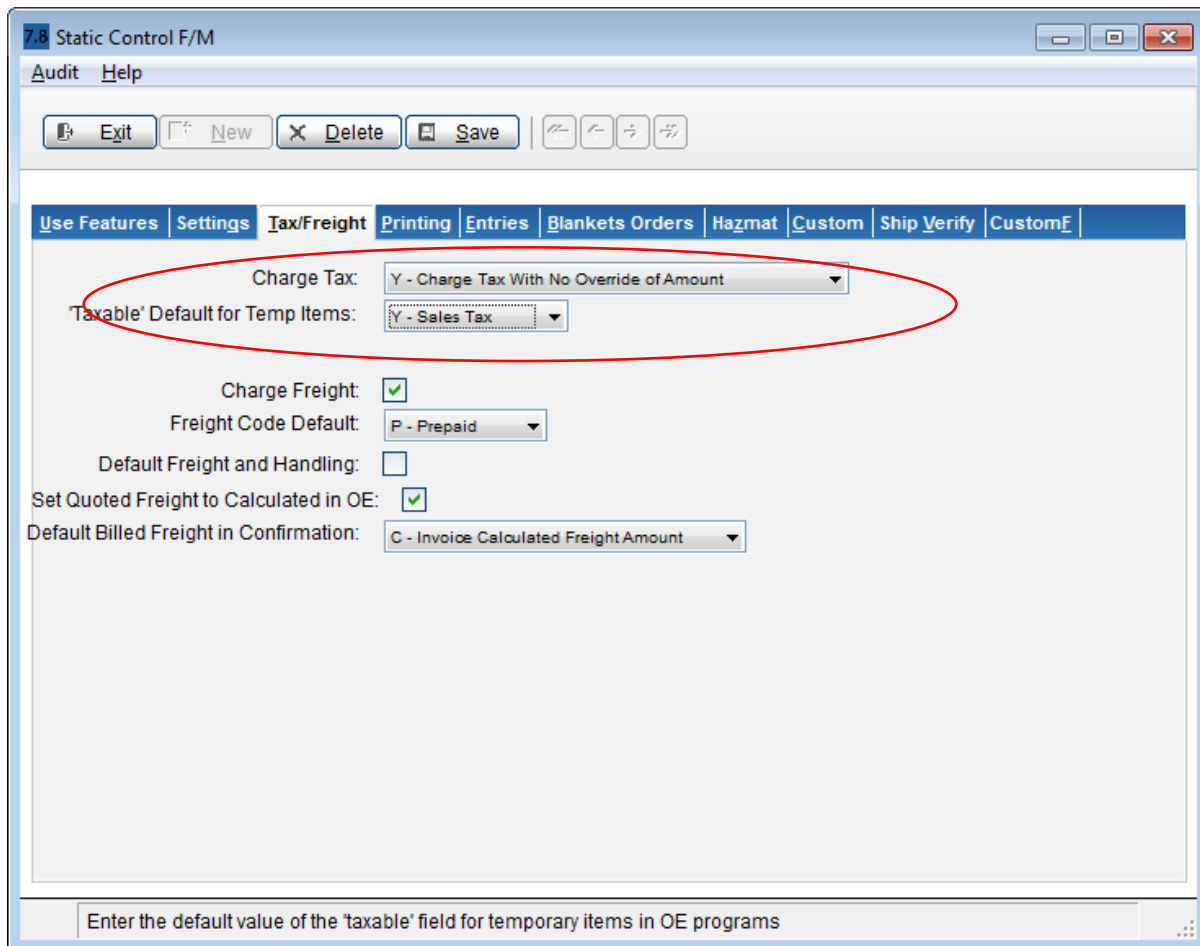


Figure 25. Sales Order Static Control F/M tax setting.

Ensure that the Tax input has "Y-Charge Tax With No Override Of Amount" selected. The connector should be in full control of all tax liability related questions. For temporary items, you may choose either "Y-Sales Tax" or "C-Cost Use Tax"

The screenshot shows a software window titled "Invoice Processing Control F/M (ARF982) - Infor, 23-KEVEN". The window has a menu bar with "Audit" and "Help". The main area contains several settings:

- "Last Invoice # Used" is a text box containing "000315".
- "Charge Tax" is a dropdown menu with "L - Charge Tax By Line" selected. This dropdown is circled in red.
- "Charge Freight" is a checkbox that is checked.
- "Allow Discount" is a dropdown menu with "% - Invoice Discount By Percent" selected.
- "Enter Cost" is a checked checkbox.
- "Post to SA" is a checked checkbox.
- "Use Department" is a checked checkbox.

Below these settings is a section titled "Invoice Print" with the following options:

- "Invoice Print" is checked.
- "Preprinted Form" is checked.
- "Address" is checked.
- "Cost Tear-Off" is unchecked.
- "Form Depth" is a text box containing "66".
- "Standard Memo" is a text box containing "THANK YOU FOR YOUR ORDER AND THIS LONG MEMO LINE!!".

At the bottom of the window are four buttons: "Save", "Delete", "New", and "Exit". A status bar at the very bottom contains the text "Enter last invoice # used".

Figure 26. AR Invoice Processing F/M Charge Tax setting.

Ensure that "Charge Tax By Line" is selected for the Charge tax input.

## Maintenance Options

There are several files that have been changed or have new implications due to implementing the FACTS AvaTax Connector. They are:

Customer File Maintenance

Item File Maintenance

Item Class File Maintenance

Adjustment Code File Maintenance

Ship Via File Maintenance

Service Code File Maintenance (if using Service & Repair module)

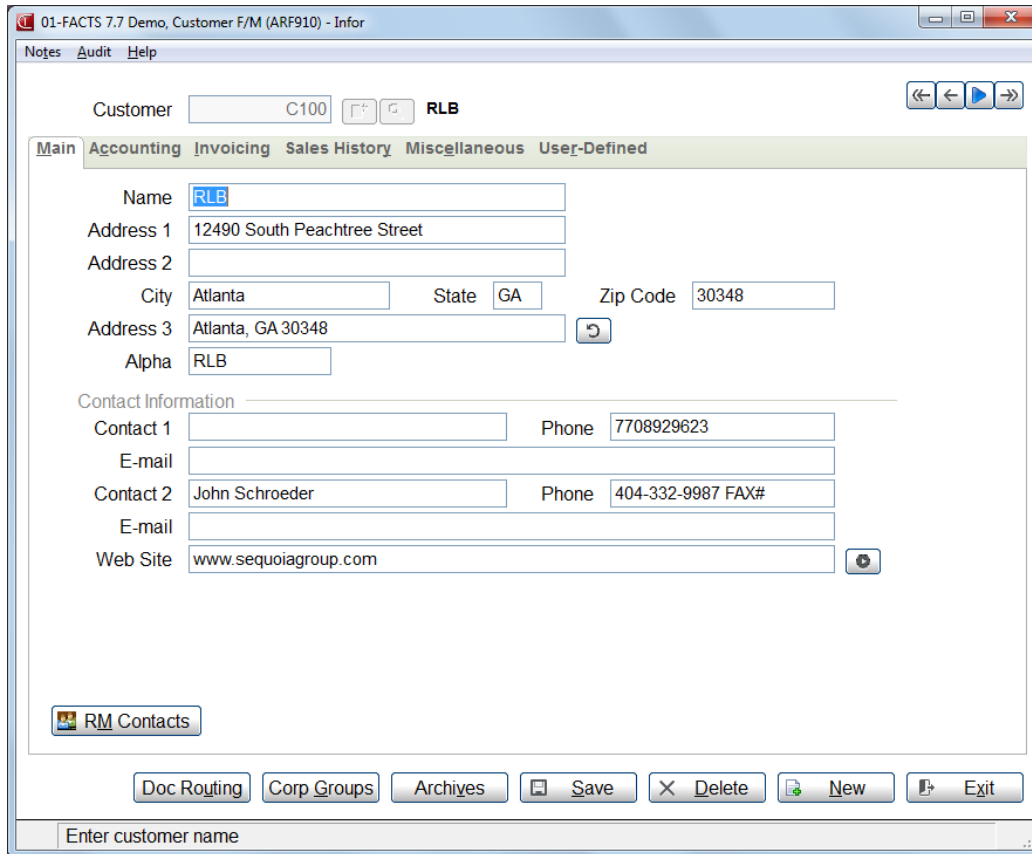


Figure 25. Customer File Maintenance Main tab

**Note for FACTS 7.7 systems:** Because there is no "Country" field, a value of "XX" in the "State" field assumes an international address. Due to the 15 character field limit with "City", address verification may attempt to truncate certain words. You may use City Correction F/M to override that behavior.

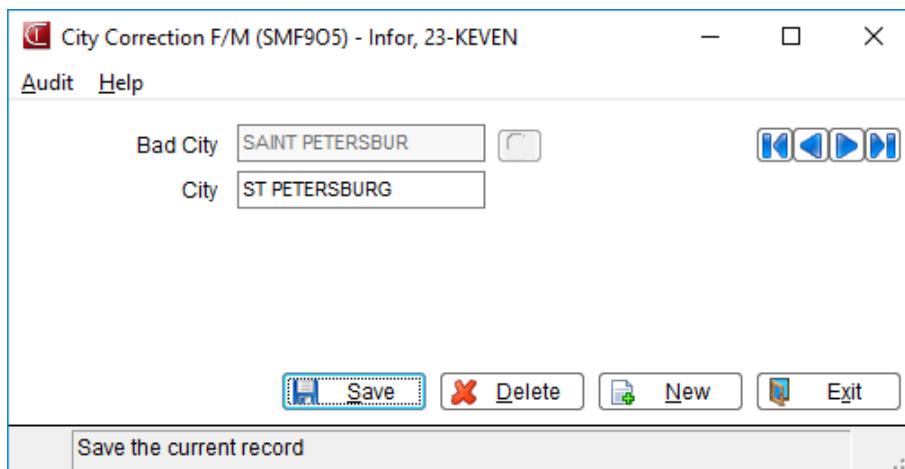


Figure 27. City Correction File Maintenance Main tab

Refer to the AvaTax Control F/M section beginning on page 11 for details as to the optional handling of the Tax Rate flag found on the Accounting tab of Customer F/M.

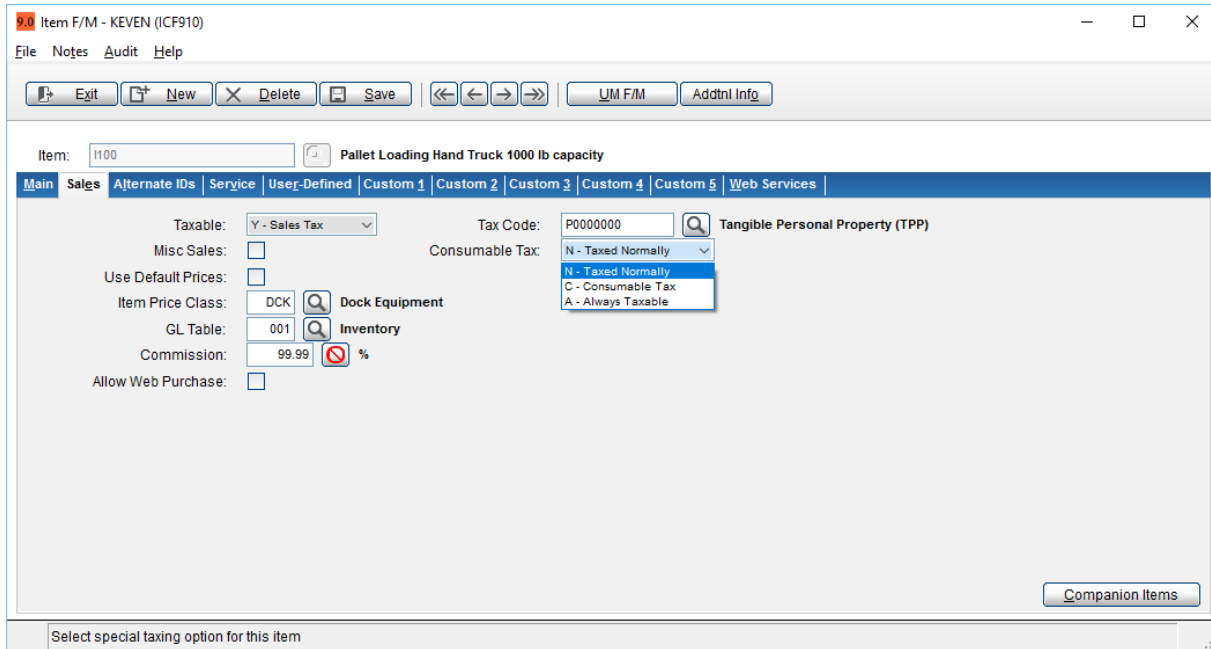


Figure 28. Item File Maintenance Sales Tab.

A new code maintained by the AvaTax Tax Code File Maintenance in the Connector Setup section determines the taxability of an item. It uses Avalara designated codes to categorize the goods or services of the item. The Taxable flag is now used to determine if Use Tax applies to this item. Consumable Tax allows for extra flexibility for determining when an item may be taxable even for customers who are normally exempt. An example would be tools or gloves that are used by the customer instead of being resold.

### Item File Maintenance Field Definitions

Field Name	Definition	Valid Values
<b>Taxable</b>	Determines whether tax is sales or use tax	Y-Sales Tax F-Force Tax C-Use Tax
<b>Tax Code</b>	Avalara tax code used to categorize the goods or services.	Any valid Avalara tax code (note: overrides "Legacy Exemption" setting)
<b>Consumable Tax</b>	Provides the ability to force taxability for customers or ship-tos whose entity code is exempt	N=Taxed Normally (default) C=Taxable when "Consumable" Tax is checked in Customer/Item Configuration A-Always Taxable

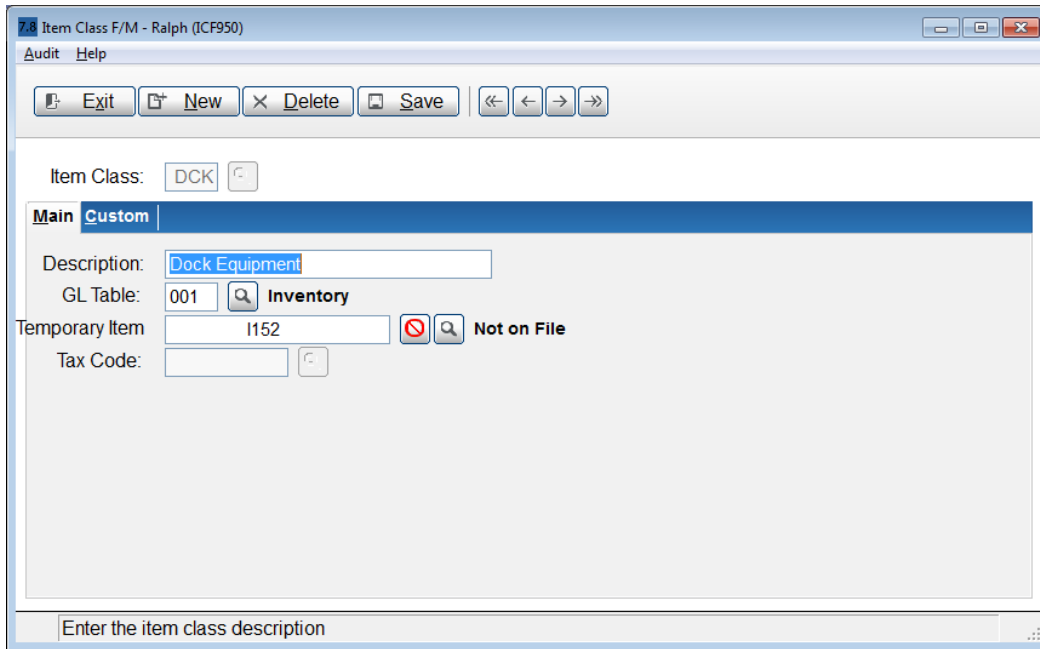


Figure 29. Item Class File Maintenance Custom Tab

Setting the AvaTax Tax Code at the Item Class level allows you to more quickly categorize the items. It also allows for item specific exceptions. If you enter a value at the item level, it will override a value assigned to its Item Class.

### Item Class File Maintenance Field Definitions

Field Name	Definition	Valid Values
<b>Tax Code</b>	Avalara tax code used to categorize the goods or services.	Any valid Avalara tax code

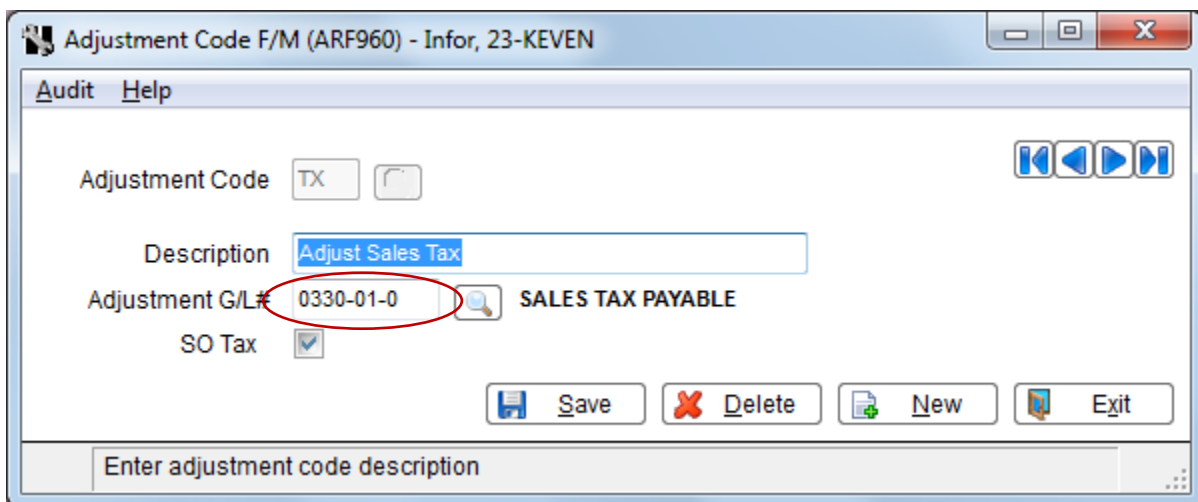


Figure 31. Adjustment Code File Maintenance

One additional field has been added to the Adjustment Code F/M to designate that a given adjustment is sales tax related. This is used in Cash Receipts and Adjustment Entry for sales tax related corrections. The use of a tax related code triggers an additional screen where specific corrections are made.

### Adjustment Code File Maintenance Added Field Definition

Field Name	Definition	Valid Values
<b>SO Tax</b>	New field to indicate whether or not adjustment is tax related	Y or N (default=N)

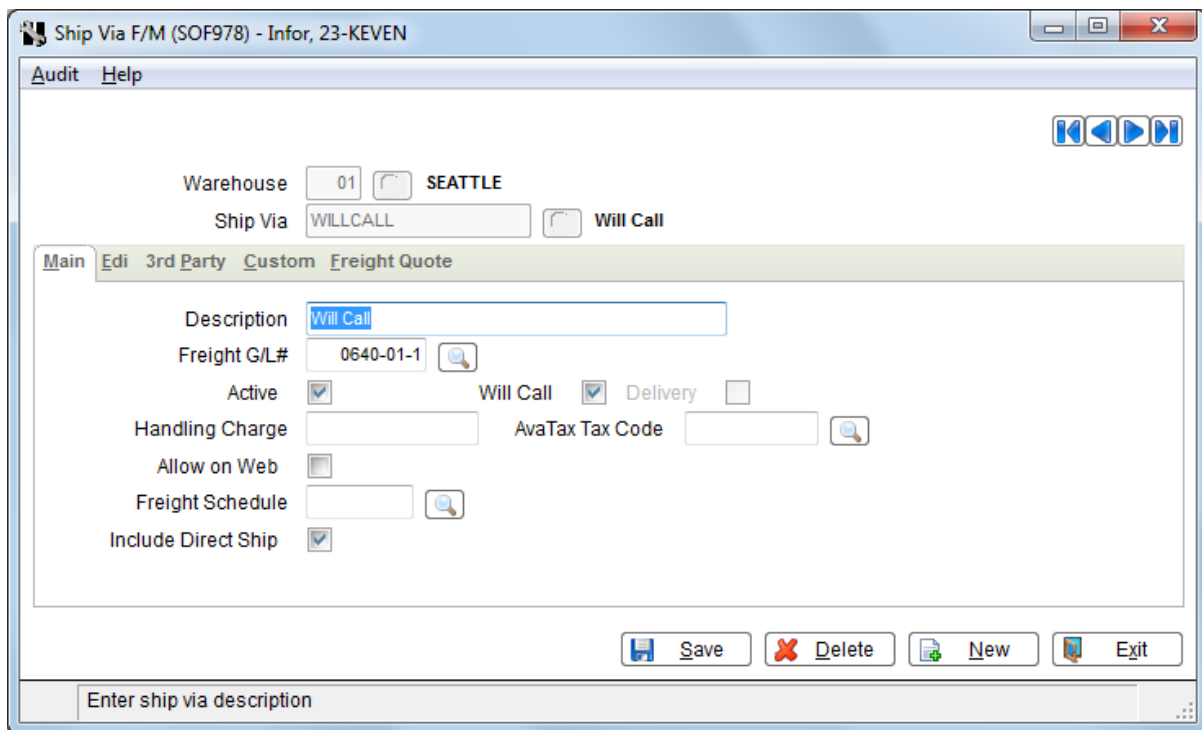


Figure 32. Ship Via File Maintenance

Two additional fields have been added to the Ship Via F/M to designate whether or not this is a Will Call pickup and the AvaTax Tax Code to be used with this method of shipment. If no code is set, the default value used is: FR020100 "Shipping only common carrier - FOB destination". If a Handling Charge is set, the default value changes to: FR030000 "Shipping And Handling Combined"

### Ship Via File Maintenance Added Field Definitions

Field Name	Definition	Valid Values
<b>Will Call</b>	New field to indicate whether or not this is a Will Call pickup. It	Y or N (default=N)

	forces the ship destination to be the same as the origin.	
<b>AvaTax Tax Code</b>	Avalara tax code used to categorize the method of shipment	Any valid Avalara tax code

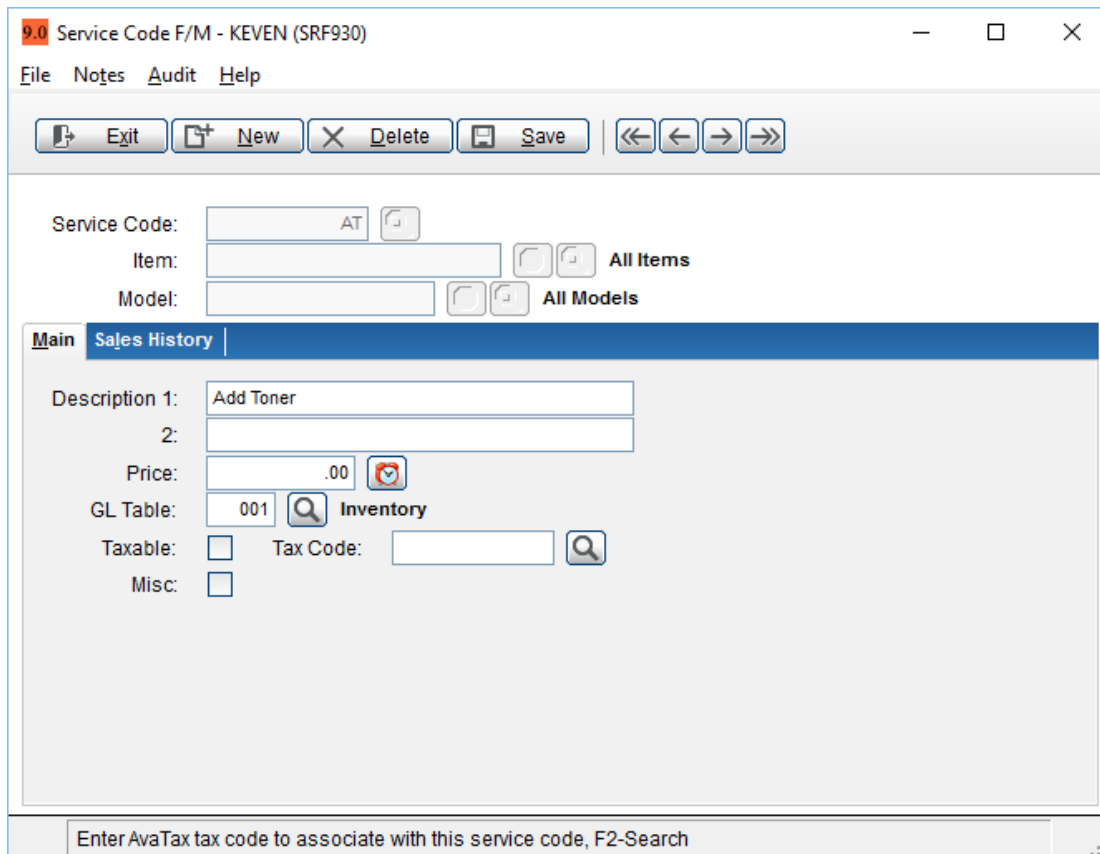


Figure 33. Service Code File Maintenance

An additional field has been added to the Service Code F/M to designate the AvaTax Tax Code to be used with service code.

**Note:** This feature is not currently supported at 7.7 and 7.9

### Service Code File Maintenance Added Field Definition

Field Name	Definition	Valid Values
<b>AvaTax Tax Code</b>	Avalara tax code used to categorize the goods or services	Any valid Avalara tax code



### Cash Receipts and Adjustment Entry Options

A unique feature of the FACTS AvaTax Connector is the ability to use Cash Receipts and Adjustment Entry to make any necessary adjustments to the tax amount of invoices. If a customer takes advantage of a terms based discount and short pays the invoice, then the tax liability is recalculated based upon the new value of the document otherwise it might be overstated. AVALARA includes a discount field in the calculation set so that amount is evaluated against tax jurisdiction based rules as to whether or not the early pay discount affects taxability. A miscalculation may also arise because one or more tax related data elements of the document content were not properly entered or configured at the time the invoice was originally generated.

Invoice Date

Customer/Entity Code

Origin or Destination

Item Tax Code

Freight Tax Code

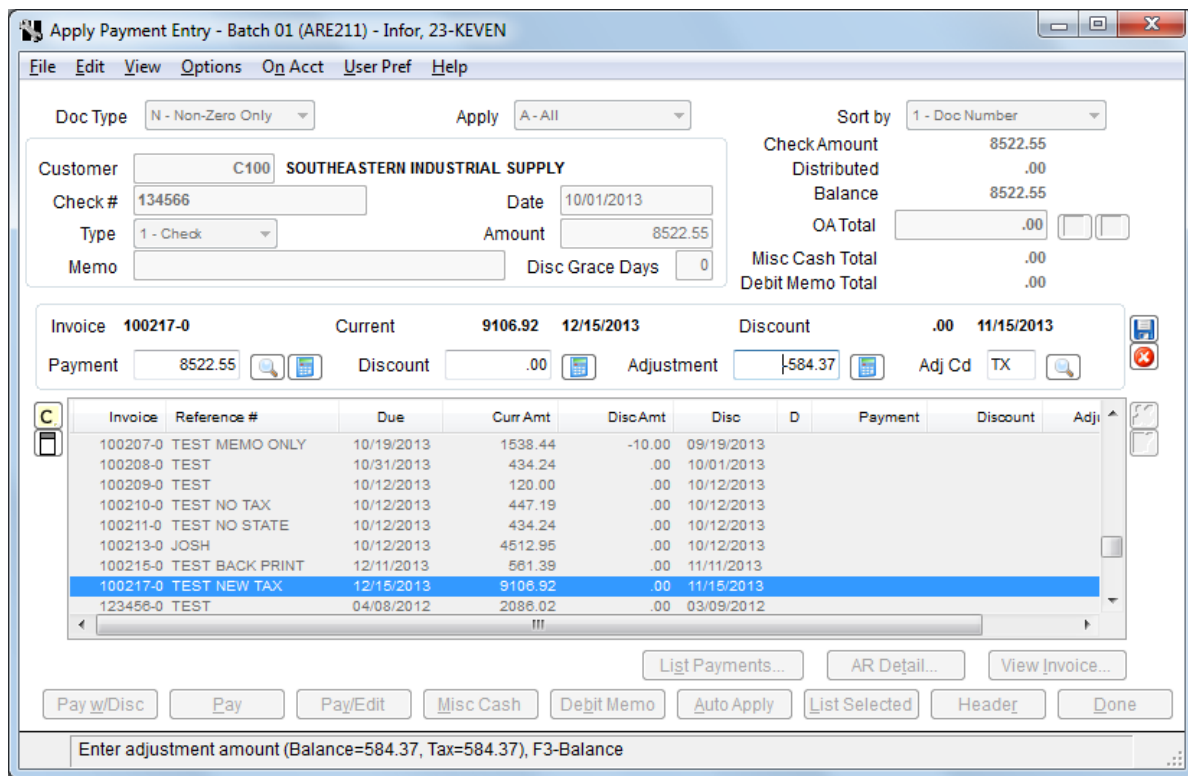


Figure 34. Cash Receipts and Adjustment Entry Payment Screen

## Cash Receipts and Adjustment Entry Payment Screen

Enter the appropriate payment amount, a discount amount if applicable and an adjustment amount if applicable. If an adjustment is made using a tax related adjustment code (see Figure 21), the Tax Adjustment Entry is triggered. You now may correct any tax related aspect of the document at either the header or line level that caused the original miscalculation.

9.3 Tax Adjustments (TXE310) - 23 KEVEN (TOK)

Help

Done

Customer: C100 Order #: 001521 Tax Calc Date: 11/15/2020  
 SOUTHEASTERN INDUSTRIAL SUPPLY Invoice #: 001724 Entity Use Code: No exempt code

Original Tax: .00 Entered Adjustment: 1.00  
 New Tax: .00 Actual Adjustment: .00  
 Total Changes: .00 Difference: 1.00  
 EarlyPay Credit: .00

Recalculate Taxes

Lines

Item: I113 ROLLING STOCK PICKING LADDER 11 STEPS. 500I  Discounts Can Apply To This Line  
 Taxable: Y - Sales Tax Tax Code: P0000000 Tangible Personal Propert  
 Origin: WH01 Destination: ST Entity Use Code: No exempt code

Item	Origin	Destination	Tax Code	Taxable	Entity Code	Disc	Description
I113	WH01	ST	P0000000	Y		Y	ROLLING STOCK PICKING LADDER 11 STEPS...
I144	WH01	ST	P0000000	Y		Y	PORTABLE SCAFFOLD SYSTEM 2 5" X 6L X 6H

Figure 35. Cash Receipts and Adjustment Entry Tax Adjustment Screen

## Cash Receipts and Adjustment Tax Adjustments Screen

Once you have completed the corrections, click the "Recalculate Taxes" button. You must fully distribute the adjustment amount to tax related changes to the document. When the Cash Receipts and Adjustment Entry register is updated, the connector will create two tax related entries that are uploaded to AvaTax: a reversal to the original record and a new one reflecting the adjustments. This saves you the time of having to credit and rebill the original order.

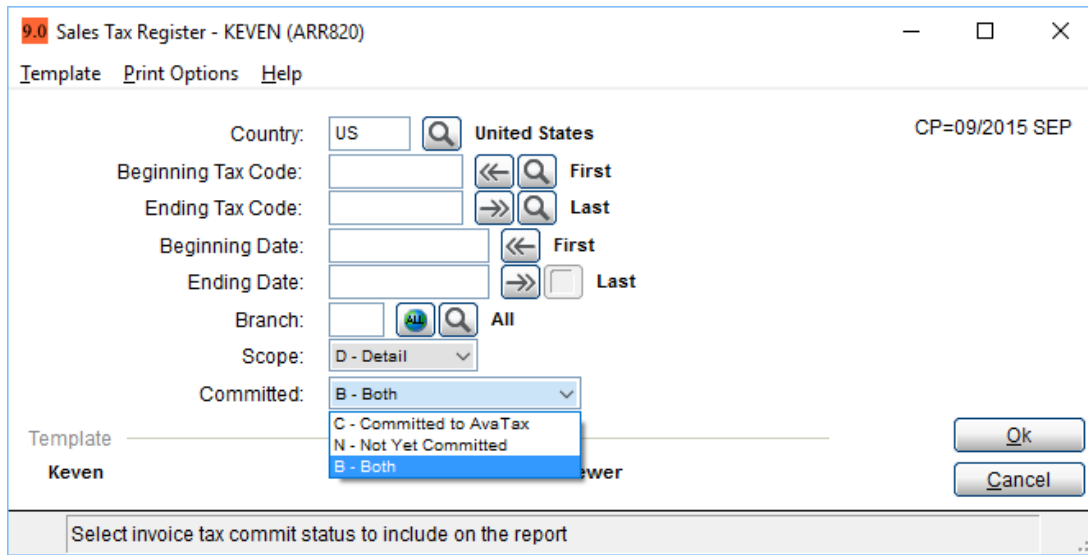


Figure 36. Sales Tax Register Screen

The FACTS Sales Tax Register continues to provide the ability to print transactions at the summary or detailed level and filtered by different criteria including country, tax code data, branch and scope. A new option allows you to filter if the invoice has been committed to the Avalara console. This is useful for those using Cash Based reporting to provide a schedule of those invoices awaiting payment that have not as yet been committed. The removal option is only available when Committed Invoices are chosen. **Note: Committed option introduced at 7.9.**

### Sales Tax Register Added Field Definition

Field Name	Definition	Valid Values
<b>Committed</b>	Determines which invoices are included based upon committed to AvaTax status	C-Committed to AvaTax N-Not Yet Committed B-Both

### Quote Processing

To facilitate quotes using "NEW" as the customer number, the "X" exempt flag will be honored even when the "Use Legacy Exempt Flags" option has not been selected using a simple addition of code found at the end of Appendix E. This is to support those using ECM, but don't want to require prospects submitting a certificate of exemption until they become a live customer.

## Order Processing

During order entry FACTS will connect with the AvaTax service to retrieve the current tax rates. No transactions will appear in the portal as a result of this process. For handling Use Tax, the line item detail screen offers a new option for the taxable flag. You select between Y-Taxable, F-Force Tax and C-Cost Use Tax. Cost Use Tax is used when an item you previously purchased tax exempt is now taxable as having been used in a billable project. Additionally, you may always force tax if necessary.

Starting with release 7.8, orders containing line items with negative quantities, i.e. items returned during a sale, reference "Usage Date" to determine taxability of when that item was originally invoiced. This ensures that the calculated tax mirrors the originally billed amount.

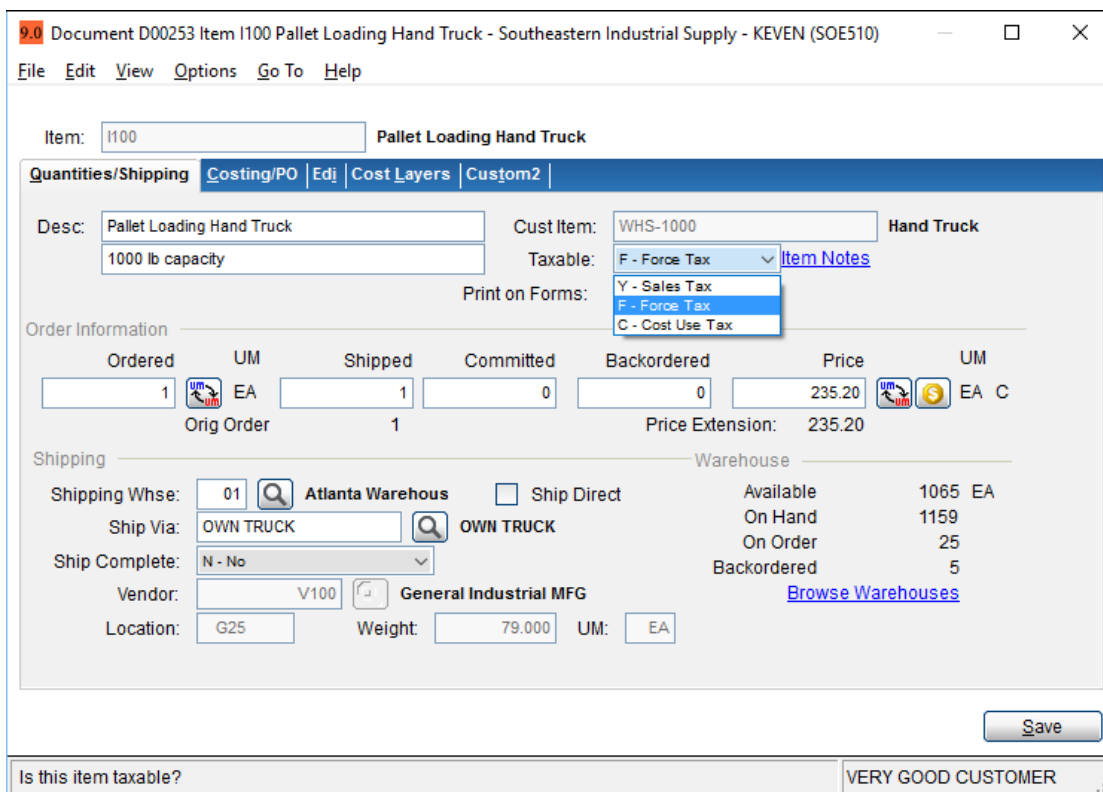


Figure 37. Sales Order Entry Detail Line Entry for Use Tax

## Order Confirmation

During order confirmation, FACTS will connect with the AvaTax service and retrieve the current tax rates and amounts if there are any tax related changes.

## Day End Processing

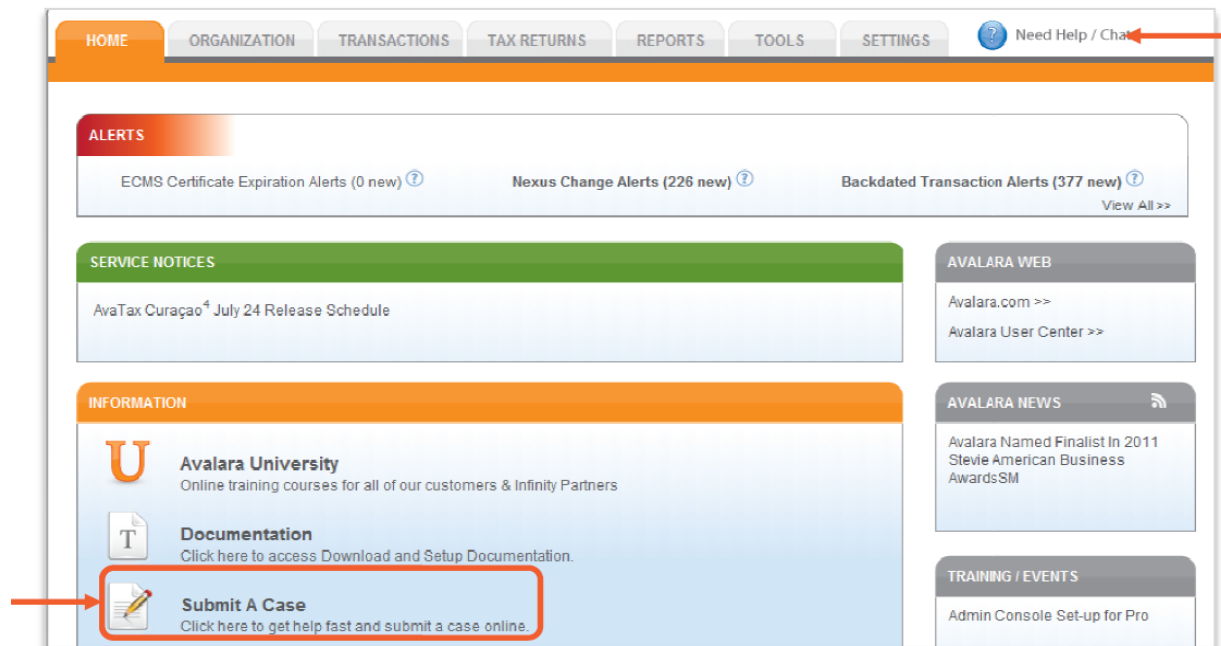
Day end processing is completely automated if you chose the Auto Update option in AvaTax Control File Maintenance. If not, you will need to manually run the Commit Invoices to AvaTax update as the final step in tax posting. Any adjustments to the transaction afterwards must be handled via Cash Receipts Entry and Adjustments or the traditional credit and rebill process.

## APPENDIX A

### Avalara Support

If you have questions about AvaTax, please contact Avalara Support with our toll free number at 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 the AvaTax Admin Console (<https://adminavatax.avalara.net/>).



The Home Screen of the AVATax Admin Console can be used to submit a Support Case.

## APPENDIX B

### Error Messages

Error. Company Not Found. Verify the Company Code. XX  
Tax Calculation Failed.

This indicates that company "XX" is not the company code used in the AvaTax web console. The company code used must exist as a valid company code in the AvaTax console.

Communication Status: FAILED: HTTP status '404' received retrieving file.

Most likely, this indicates you have made an error with the Avalara URL in the AvaTax Control F/M.

An http status 404 uploading customers is usually an indication that the customer or ship-to address is incomplete and likely missing the state.

## APPENDIX C

### Generic Tax Calculation

For developers who wish to leverage the FACTS Connector for other PXPlus based applications, the self documenting code is found below.

```

34900 AVATAX_GENERIC_CALC:! ^100,5
34904 ! Note that this routine will calculate tax, but it will never commit to AvaTax, so none of these will
ever be reportable tax events.
34905 ! Call/Enter variables are as follows:
34906 ! arcust = channel number for ARCUST data file - optional.
34907 ! arship = channel number for ARSHIP data file - optional.
34908 ! The next variable is 'options$' and is being set by the tbl commands.
34909 ! doc_num$ is required. It is a unique identifier. It does not need to be unique if you don't care
about logs, etc. but it cannot be blank. Recommended 6 digits.
34910 ! cust_num$ = customer number - required and must be a valid customer number.
34911 ! doc_date$ = date for affective tax calculation - required. Six digit packed julian date.
34912 ! tax_rate_flag$ = document tax rate flag H, L, or X - optional, will default to H if left blank.
34913 ! ship_to_num$ = Customer's ship-to - optional, will default to " SAME" if left blank.
34914 ! reference_num$ = Reference Number - optional.
34915 ! branch$ = Branch where the transaction occurred - optional IF whse$ is supplied.
34916 ! whse$ = Warehouse where the transaction occurred - optional IF branch$ is supplied.
34917 ! ship_via$ = Ship Via for this transaction - optional. Ship Via will only be used if whse$ is supplied.
Ship Via will be used to see if the transaction should be taxed as a will call.
34918 ! tax_code$ = AvaTax tax code normally assigned to an item - optional.
34919 ! entity_use_code$ = AvaTax entity use code - optional. If not supplied, entity use code assigned to
customer, ship-to, etc. will still be looked up and used.
34920 ! line_tax_flag$ = line item tax flag we would normally get from SORSOL - optional. If not supplied
'Y' will be assumed.
34921 ! quantity$ = quantity to multiply by price before calculating tax - optional. If not supplied 1 is
assumed.
34922 ! price$ = price to multiply by quantity before calculating tax - required. This can be the full
amount to calculate tax on if quantity$ is blank or set to 1.
34923 ! invoice_discount$ = Discount amount to pass into the tax calc engine (if any) - optional. This is
the same as the invoice discount on the footer of a sales order invoice.
34925 ! doc_taxable is returned from the tax calc and will be the taxable amount.
34926 ! doc_total_tax is returned from the tax calc and will be the amount of tax calculated.
34927 ! doc_tax_rate is returned from the tax calc and is the net combined tax rate.
34928 ! tax_status$ = status of transaction returned from the tax calc. OK=no problems.
34930 call
"prog/TX/TXC010;single_amount_tax",(arcust),(arship),tbl(get_avatax_detail,"","line_detail,")+tbl(_no_ui,"","s
ilent"),(doc_num$),(cust_num$),(doc_date$),(tax_rate_flag$),(ship_to_num$),(reference_num$),(branch$),(w
hse$),(ship_via$),(tax_code$),(entity_use_code$),(line_tax_flag$),(quantity$),(price$),num(invoice_discount$)
,doc_taxable,doc_total_tax,doc_tax_rate,tax_status$
34945 if tax_status$ <> "OK" and tax_status$ <> "Tax server is inactivated in Tax Static Control F/M" then
z$="Tax calculation failed"; gosub 8810
34995 return

```



## APPENDIX D

### Features by FACTS Version – initially released at 7.7 and retrofitted to 7.5

- Returned items (negative quantity) in sales orders referencing Usage Date for calcs – 7.8
- Avalara company code in Branch F/M – 7.8
- Cash Basis calculations – 7.8
- Find Obsolete Tax Codes – 7.8
- Import Items/Classes from Previous Customer in Customer/Item Configuration – 7.9
- Post Past Invoices (not including Cash Basis) - 7.9
- Update Past Invoices – 9.1
- Use Register Date – 9.1
- Geolocation via entry of latitude and longitude in Customer/Item Configuration – 9.1
- Cash Basis support using Post Past Invoices – 9.2
- Statistics tab in Control F/M showing “hits” activity – 9.3
- Ability to exempt “NEW” customer in quote entry while not using legacy exempt flags – 9.3
- Options by State F/M for controls specific to individual states – 9.3

## APPENDIX E

### Partner Custom Modification Hook Points

The FACTS Connector contains a number of hook points enabling partners to introduce modifications. There is also a variable "custom\_avatax\_options\$" that can be loaded with custom data for use in these routines. Each data piece in this string needs to end with a comma. Here are two examples:

```
custom_avatax_options$="mycustomoption," or  
custom_avatax_options$="mycustomoption,customfrt=12.95,"
```

Below is a list of the hook points found in the tax calculation engine TXC010 that call out to a custom overlay. Note that TXC010 is a protected program. We will be happy to work with your technical resources to fully explain how to leverage these hook points.

```
2227 perform "prog/TX/TXC01A;init_soquote_docs",err=*next
```

This is performed when tax is to be calculated for a sales order quote. It is performed after we have completed our standard initialization including opening any files channels, but before the quote information is gathered and assembled to be sent to AvaTax.

```
2337 perform "prog/TX/TXC01A;init_so_order_docs",err=*next  
Same as above, but for sales order orders and invoices.
```

```
2753 perform "prog/TX/TXC01A;init_sr_invoice_docs",err=*next  
in the initialization routine for service and repair invoice tax calc.
```

```
3127 perform "prog/TX/TXC01A;init_txacp_docs",err=*next  
in the initialization routine for cash receipts tax adjustments.
```

```
3737 perform "prog/TX/TXC01A;init_past_invc_docs",err=*next
```

This is the same as above. There is a routine that calculates tax on a past invoice. It is used when tax needs to be adjusted on an invoice that was completed before AvaTax was in use. It give us an AvaTax answer to work against when recording an adjustment of an invoice to AvaTax. This hook point is in the initialization section of this routine.

```
5385 perform "prog/TX/TXC01A;exit_pgm",err=*next
```

This routine is performed after file channels are closed and before TXC010 is exited.

```
20940 if not(checked_txc01a_read_arcust) or got_txc01a_read_arcust then  
checked_txc01a_read_arcust=1; perform "prog/TX/TXC01A;read_arcust",err=*next;  
got_txc01a_read_arcust=1
```

There is a block of code call 'read\_arcust' that all of the routines in TXC010 use whenever it needs to get customer information. This routine reads arcust\$ from arcust. It also reads TXEUCH, looking for a record for the specific customer. This perform happens after both of these reads have finished.

```
20980 if not(checked_txc01a_read_arship) or got_txc01a_read_arship then
checked_txc01a_read_arship=1; perform "prog/TX/TXC01A;read_arship",err=*next;
got_txc01a_read_arship=1
```

Same as above, but for a customer ship-to. By the time it does this perform, it has read arship into arship\$ and txeuch into txeuch\$ for the specific customer and ship-to. If the read to arship is not successful, then the routine is exited before this perform takes place.

```
21020 if not(checked_txc01a_read_apvend) or got_txc01a_read_apvend then
checked_txc01a_read_apvend=1; perform "prog/TX/TXC01A;read_apvend",err=*next;
got_txc01a_read_apvend=1
```

This routine is performed after apvend is read. However, the read\_apvend block currently does not read the entire apvend record. It reads in vendor\_city\$, vendor\_zip\_code\$, vendor\_addr1\$, vendor\_addr2\$, vendor\_addr3\$, vendor\_state\$, vendor\_country\$, and vendor\_addr2\$

```
21070 if not(checked_txc01a_read_icwhcd) or got_txc01a_read_icwhcd then
checked_txc01a_read_icwhcd=1; perform "prog/TX/TXC01A;read_icwhcd",err=*next;
got_txc01a_read_icwhcd=1
```

This perform happens after icwhcd has been read. The entire icwhcd record is not read. The variables it does read in are whse\_addr1\$, whse\_addr2\$, whse\_addr3\$, whse\_city\$, whse\_state\$, whse\_zip\_code\$, whse\_country\$, and whse\_addr2a\$

```
24212 perform "prog/TX/TXC01A;get_exclude_states",err=*next
```

There is a block of code in TXC010 called filter\_states. This routine is used before the tax request is built and sent to AvaTax. In this hook point, any 2 digit state codes added to exclude\_states\$ will be used to filter against. If a 'destination state' is found in exclude\_states\$ then the related transaction will not be sent to AvaTax.

```
30630 perform "prog/TX/TXC01A;post_get_soqsoh_header_info",err=*next
```

This perform is at the end of the get\_soqsoh\_header\_info block. This block of code does a gosub read\_arcust (if customer is not NEW), and a gosub to read arship (if ship-to is not SAME). The main thing this block of code does is sets the variables arship.ship\_to\_name, arship.ship\_to\_addr\_1\$, arship.ship\_to\_addr\_2\$, etc. These variables are set by reading arship, or if ship-to is SAME, these variables are set to their corresponding arcust counterparts. If

customer is 'NEW' then these variables are set based on the address information in SOQSOH. So, this hook point is designed to allow you to change ship-to address information for a quote if needed.

```
30877 perform "prog/TX/TXC01A;post_get_quote_lines",err=*next
```

This hook point is in the get\_quote\_lines routine which is responsible for getting quote line information and writing to a memory file. The memory file is later read when building the xml for AvaTax. If there are freight charges on the quote, this is also written to the line item memory file. Here are the writes to the memory file in this block:

```
30850 write
```

```
(doc_lines)soqsol.line_num$,destination_address_code$,origin_address_code$,stp(soqsol.item_num$,2),tax_code$,stp(entity_use_code$,2),stp(soqsol.description_1$,2)+tbl(nul(soqsol.description_1$) or nul(soqsol.description_2$),"
```

```
",") +stp(soqsol.description_2$,2),str(abs(num(soqsol.quantity$))),line_extension$,discountable$
```

```
30870 write
```

```
(doc_lines)"FRT",tbl(will_call$="Y","ST","WH"+init_whse$),"WH"+init_whse$,"Freight",tbl(nul(ship_via_tax_code$),stp(ship_via_tax_code$,2),"FR000000"),stp(entity_use_code$,2),"Outbound Freight","1",str(doc_freight),"
```

Using the hook point on line 30877, you can remove, manipulate, or add line item information to the memory file as needed.

```
30980 perform "prog/TX/TXC01A;post_get_sorsoh_header_info",err=*next
```

See explanation of post\_get\_soqsoh\_header\_info above since this is pretty much the same, other than there is no 'NEW' customer option in sorsoh. Also, this get\_sorsoh\_header\_info block of code also sets tax\_date\$ and doc\_date\$ based on the following: Invoice or counter sale, doc\_date\$=invc\_date\$; credit memo, set based on date selection in tax control. Orders do not have an invc\_date\$ but order\_date\$ is set to the order date. So, you can change these dates if needed using this hook point as well.

```
31257 if not(checked_so_ship_pre_write_doc_line) or got_so_ship_pre_write_doc_line then
```

```
checked_so_ship_pre_write_doc_line=1; perform
```

```
"prog/TX/TXC01A;so_ship_pre_write_doc_line",err=*next; got_so_ship_pre_write_doc_line=1
```

This hook point happens in the get\_order\_lines block right before a sales order document line is written to the line item memory file as described above (for post\_get\_quote\_lines). Here is the write to the memory file that happens right after this perform is attempted:

```
31260 write
```

```
(doc_lines)sorsol.line_num$,destination_address_code$,origin_address_code$,stp(sorsol.item_num$,2),tax_code$,stp(entity_use_code$,2),stp(sorsol.description_1$,2)+tbl(nul(sorsol.description_1$) or nul(sorsol.description_2$),"
```

```
",") +stp(sorsol.description_2$,2),str(abs(num(line_quantity$))),line_extension$,discountable$
```

Note that if line\_extension\$ is zero then the line will not be written to the memory file.

```
31310 perform "prog/TX/TXC01A;post_get_order_lines",err=*next
```

See explanation of post\_get\_quote\_lines. This is in the same get\_order\_lines block as the above hook point but happens at the end of the block after lines and freight has been written to the memory file. Here is the logic that writes the freight to the memory file:

```
31280 select junk$,*,frt_whse$,*,freight$,*,clippership$ from sofrwh,kno=0 begin
%a0$+doc_num$ end %a0$+doc_num$+%lc$ where
num(freight$)+num(mid(clippership$,14,11))<>0
31285 doc_freight=num(freight$)+num(mid(clippership$,14,11))
31290 if doc_freight<>0 then write
(doc_lines)"FRT"+frt_whse$,tbl(will_call$="Y","ST","WH"+frt_whse$),"WH"+frt_whse$,"Freight",tbl
(nul(ship_via_tax_code$),stp(ship_via_tax_code$,2),tbl(num(mid(clippership$,14,11))<>0,"FR0000
00","FR030000")),stp(entity_use_code$,2),"Outbound Freight","1",str(doc_freight),"
31295 next record
```

```
31312 TXC01A;so_ship_pre_write_freight_line
```

This is inside of the SOFRWH loop where it is pulling freight record(s) to add to the order when submitting to AvaTax. It was added to the sales order block and the service and repair document block. (the same as TXC01A;so\_ship\_pre\_write\_doc\_line)

```
31457 if not(checked_so_back_pre_write_doc_line) or got_so_back_pre_write_doc_line then
checked_so_back_pre_write_doc_line=1; perform
"prog/TX/TXC01A;so_back_pre_write_doc_line",err=*next; got_so_back_pre_write_doc_line=1
```

This is the same as the so\_ship\_pre\_write\_doc\_line, but happens for backordered lines. Here is the write to the memory file that happens after this perform:

```
31460 write
(doc_lines)sorsol.line_num$,destination_address_code$,origin_address_code$,stp(sorsol.item_nu
m$,2),tax_code$,stp(entity_use_code$,2),stp(sorsol.description_1$,2)+tbl(nul(sorsol.description_1
$) or nul(sorsol.description_2$),"
","")+stp(sorsol.description_2$,2),str(abs(line_quantity)),line_extension$,discountable$
```

```
32248 if not(checked_so_ship_pre_write_doc_line) or got_so_ship_pre_write_doc_line then
checked_so_ship_pre_write_doc_line=1; perform
"prog/TX/TXC01A;so_ship_pre_write_doc_line",err=*next; got_so_ship_pre_write_doc_line=1
```

This hook point is in the get\_sr\_backorder\_lines (related to service and repair). It is in the block of code that is writing sorsol lines to the memory file (if mid(w7\$,1,1)+mid(w7\$,46,1)<>"SA"). Here is the write to the memory file that happens after this hook point.

32251 write

```
(doc_lines)sorsol.line_num$,destination_address_code$,origin_address_code$,stp(sorsol.item_num$,2),tax_code$,stp(entity_use_code$,2),stp(sorsol.description_1$,2)+tbl(nul(sorsol.description_1$) or nul(sorsol.description_2$),"
,"")+stp(sorsol.description_2$,2),str(abs(num(line_quantity$))),line_extension$,discountable$
```

32448 if not(checked\_so\_ship\_pre\_write\_doc\_line) or got\_so\_ship\_pre\_write\_doc\_line then  
checked\_so\_ship\_pre\_write\_doc\_line=1; perform

```
"prog/TX/TXC01A;so_ship_pre_write_doc_line",err=*next; got_so_ship_pre_write_doc_line=1
```

Same as above, but for non-backordered service and repair lines. Here is the write to the memory file that happens after this hook point.

32451 write

```
(doc_lines)sorsol.line_num$,destination_address_code$,origin_address_code$,stp(sorsol.item_num$,2),tax_code$,stp(entity_use_code$,2),stp(sorsol.description_1$,2)+tbl(nul(sorsol.description_1$) or nul(sorsol.description_2$),"
,"")+stp(sorsol.description_2$,2),str(abs(num(line_quantity$))),line_extension$,discountable$
```

34880 perform "prog/TX/TXC01A;post\_get\_sopinh\_header\_info",err=\*next

When calculating taxes for a past invoice that was completed before AvaTax was in use, there is a get\_sopinh\_header\_info routine that is used to get header information. This block of code set arship.ship\_to\_addr\_1\$, arship.ship\_to\_addr\_2\$, etc. based on information found in sopina. It also conditionally sets doc\_date\$ and invc\_date\$ as described above. So, you can change these values as needed with this hook point.

35057 if not(checked\_so\_past\_pre\_write\_doc\_line) or got\_so\_past\_pre\_write\_doc\_line then  
checked\_so\_past\_pre\_write\_doc\_line=1; perform

```
"prog/TX/TXC01A;so_past_pre_write_doc_line",err=*next; got_so_past_pre_write_doc_line=1
```

Same as so\_ship\_pre\_write\_doc\_line described above, except for past invoices. Here is the write to the memory file that happens right after this line of code:

35060 write

```
(doc_lines)sopind.line_num$,destination_address_code$,origin_address_code$,stp(sopind.item_num$,2),tax_code$,stp(entity_use_code$,2),stp(sopind.description_1$,2)+tbl(nul(sopind.description_1$) or nul(sopind.description_2$),"
,"")+stp(sopind.description_2$,2),str(abs(num(line_quantity$))),line_extension$,discountable$
```

35100 perform "prog/TX/TXC01A;post\_get\_past\_invc\_lines",err=\*next

Same as post\_get\_order\_lines described above, except for past invoices. Note that freight is read out of sopfrw instead of sofrwh.

35275 if not(checked\_so\_past\_pre\_write\_doc\_line) or got\_so\_past\_pre\_write\_doc\_line then  
checked\_so\_past\_pre\_write\_doc\_line=1; perform  
"prog/TX/TXC01A;so\_past\_pre\_write\_doc\_line",err=\*next; got\_so\_past\_pre\_write\_doc\_line=1  
35450 perform "prog/TX/TXC01A;post\_get\_past\_sr\_inv\_lines",err=\*next

Same as the two hook points above, but for past service and repair lines.

TXALIB is not protected so anyone can look at this program to get more information about these hook points. Most of these follow a tax calculation by AvaTax and are performed before the FACTS variables are overridden by the values returned from AvaTax.

23950 perform "prog/TX/TXALIC;get\_sales\_tax\_values",err=\*next  
24320 if not(checked\_arr214\_update\_arcshl) or got\_arr214\_update\_arcshl then  
checked\_arr214\_update\_arcshl=1; perform "prog/TX/TXALIC;arr214\_update\_arcshl",err=\*next;  
got\_arr214\_update\_arcshl=1

25720 perform "prog/TX/TXALIC;get\_entity\_use\_code",err=\*next

26350 if not(checked\_write\_txadjl) or got\_write\_txadjl then checked\_write\_txadjl=1; perform  
"prog/TX/TXALIC;write\_txadjl",err=\*next; got\_write\_txadjl=1

26950 if not(checked\_update\_sopinh\_tax) or got\_update\_sopinh\_tax then  
checked\_update\_sopinh\_tax=1; perform "prog/TX/TXALIC;update\_sopinh\_tax",err=\*next;  
got\_update\_sopinh\_tax=1  
30130 perform "prog/TX/TXALIC;avatax\_are112\_calc",err=\*next; on go\_back goto 30138,30125

30230 perform "prog/TX/TXALIC;avatax\_arp11\_calc",err=\*next; on go\_back goto 30238,30225

30330 perform "prog/TX/TXALIC;avatax\_aru111\_calc",err=\*next; on go\_back goto 30338,30325

30530 perform "prog/TX/TXALIC;avatax\_are132\_calc",err=\*next; on go\_back goto 30538,30525

30720 perform "prog/TX/TXALIC;avatax\_soc114\_calc",err=\*next; on go\_back goto 30728,30715

31025 perform "prog/TX/TXALIC;avatax\_soc490\_calc",err=\*next; on go\_back goto 31033,31020

31140 perform "prog/TX/TXALIC;avatax\_soc504\_calc",err=\*next; on go\_back goto 31148,31115

31320 perform "prog/TX/TXALIC;avatax\_soc755\_calc",err=\*next; goto 31328

31450 perform "prog/TX/TXALIC;avatax\_sop31\_calc",err=\*next; on go\_back goto 31458,31430

31520 perform "prog/TX/TXALIC;avatax\_sop32\_calc",err=\*next; on go\_back goto 31528,31515

31630 perform "prog/TX/TXALIC;avataxc\_soe352\_calc",err=\*next; on go\_back goto 31638,31625

31830 perform "prog/TX/TXALIC;avatax\_sou321\_calc",err=\*next; on go\_back goto 31838,31825

31920 perform "prog/TX/TXALIC;avatax\_sre31\_single\_line\_calc",err=\*next; on go\_back goto 31928,31915

32520 branch\_to=0; perform "prog/TX/TXALIC;avatax\_SRE512\_calc",err=\*next; on branch\_to goto \*next,32548,32515

32725 branch\_to=0; perform "prog/TX/TXALIC;avatax\_SRE332\_calc",err=\*next; on branch\_to goto \*next,32753,32720

32830 branch\_to=0; perform "prog/TX/TXALIC;avatax\_sru591\_calc",err=\*next; on branch\_to goto \*next,32858,32825

32920 perform "prog/TX/TXALIC;avatax\_erc114\_calc",err=\*next; on go\_back goto 32928,32915

33220 perform "prog/TX/TXALIC;avatax\_erp11\_calc",err=\*next; on go\_back goto 33228,33215

33320 perform "prog/TX/TXALIC;avatax\_erp15\_calc",err=\*next; on go\_back goto 33328,33315

33420 perform "prog/TX/TXALIC;avatax\_erc124\_calc",err=\*next; on go\_back goto 33430,33415

33435 perform "prog/TX/TXALIC;avatax\_erc124\_calc\_current",err=\*next; on go\_back goto 33443,33430

33620 perform "prog/TX/TXALIC;avatax\_erp12\_calc",err=\*next; on go\_back goto 33628,33615

33720 perform "prog/TX/TXALIC;avatax\_ere232\_calc",err=\*next; on go\_back goto 33728,33715

33920 perform "prog/TX/TXALIC;avatax\_erc214\_calc",err=\*next; on go\_back goto 33928,33915

33935 perform "prog/TX/TXALIC;avatax\_erc214\_calc\_current",err=\*next; on go\_back goto 33938,33930

34120 perform "prog/TX/TXALIC;avatax\_erp22\_calc",err=\*next; on go\_back goto 34128,34115

34220 perform "prog/TX/TXALIC;avatax\_erp23\_calc",err=\*next; on go\_back goto 34228,34215

34320 perform "prog/TX/TXALIC;avatax\_ere260\_calc",err=\*next; on go\_back goto 34328,34315

34335 perform "prog/TX/TXALIC;avatax\_ere260\_calc\_current",err=\*next; on go\_back goto 34338,34330



34425 perform "prog/TX/TXALIC;avatax\_ws\_calc",err=\*next; on go\_back goto 34433,34420

34640 perform "prog/TX/TXALIC;avatax\_txu141\_calc",err=\*next; on go\_back goto 34648,34615

36017 perform "prog/TX/TXALIC;init\_check\_sorsol\_delivery\_fee\_line",err=\*next

Quote Entry Force Exempt Entity Code for customer "NEW" in quote entry only:

Add the following line of code to "prog/TX/TXALIC;get\_entity\_use\_code" :

if stp(cust\_num\$,2)="NEW" and tax\_rate\_flag\$="X" then entity\_use\_code\$="L"

## APPENDIX F

### DOU100 Bug Fix

Normally, the last record in DOFILH is TUPROB1. There is special code in DOU100 that skips TU records and ends the update properly (see line 1045). This masks the fact that there is a bug in DOU100 since the bug only manifests itself if there are additional data files after the TU files. The problem is that when it hits the last record in DOFILH, if the last record is not a TU record, it stays on that last record, instead of hitting end of file and then fails when it tries to process TXUSEC the second time.

#### For 7.7 through 9.2:

Add this line:

```
10823 if k$(1,8)<>cur$(1,8) then goto FINALIZE_DD_OBJECT ! %xq.xx_system CK -- 9/11/2006 --  
Check for end of alternate keys now (before read) so we don't need to roll the key pointer back  
in FINALIZE_DD_OBJECT. This will correct a problem which happens when the last record in  
DOFILH is a normal file (not a TU file). Also see lines 16060 and 16065 which are disabled as  
they are not necessary with this line in place.
```

Remark out these lines:

```
16060 ! let K$=kep(DOFILH,end=*next) ! %xq.xx_system - see line 10823  
16065 ! read (DOFILH,key=K$,dir=0) ! %xq.xx_system - see line 10823
```

#### For 7.6 and prior:

Add this line:

```
10323 if k$(1,8)<>cur$(1,8) then goto FINALIZE_DD_OBJECT ! CK -- 9/11/2006 -- Check for end  
of alternate keys now (before read) so we don't need to roll the key pointer back in  
FINALIZE_DD_OBJECT. This will correct a problem which happens when the last record in  
DOFILH is a normal file (not a TU file). Also see lines 19045 and 19050 which are disabled as  
they are not necessary with this line in place.
```

Remark out these lines:

```
19045 ! let K$=kep(DOFILH,end=*next) ! CK -- 9/11/2006 -- This line is not necessary since the  
new line 10323 eliminates the need to roll the key back.  
19050 ! read (DOFILH,key=K$,dir=0) ! CK -- 9/11/2006 -- This line is not necessary since the new  
line 10323 eliminates the need to roll the key back.
```

**APPENDIX G**

## Software Port Assignments

http=80 (outbound)

https=443 (outbound)