CS1321 Sales Tax Computation Program*

Jeffrey D. Oldham

2000 Mar 26

1 Introduction

In most states, sales tax is charged when purchasing items at stores and sometimes for catalog or Internet sales. Currently, there are over seven thousand different tax jurisdiction, each with its own tax rate and regulations. The growth of Internet-based shopping and ecommerce promises to complicate the convoluted codes.

Recently, the National Governors' Association proposed a simplified sales tax system [Nat99]. Each merchandising company would contract with an independent third-party organization to calculate and collect sales taxes, remitting the funds to the appropriate state governments.

1.1 Proposal

We propose to construct an Internet-based sales tax system capable of calculating the appropriate sales tax on a group of taxable and tax-exempt items for any particular tax jurisdiction. When sent an address and a list of items, the server program will compute the correct amount of sales tax for the appropriate tax jurisdiction. To do so, the program must determine the jurisdiction, its set of rules for which products are taxable, match this list against the submitted items, and then compute the sales tax rate.

The design and implementation of the program will reveal which aspects of proposed sales tax "schemes" can be automated. We will also develop and justify recommendations how to simplify aspects of current sales tax systems that cannot easily be automated. If successful and our recommendations are implemented by state legislatures, we hope to reduce the cost of complying with sales tax regulations by an order of magnitude. We may also be able to spin-off our work to form a company specializing in sales tax computation and collection.

2 A Survey of the Current Sales Tax System

The current sales tax system is characterized by

- numerous tax jurisdictions, some consisting of very small towns, each having its own tax rate and many having their own sales tax rules,
- collection costs borne by retailers both when purchasing items from suppliers and when selling to the public, and
- no mandatory tax collection when selling across state lines.

2.1 Numerous Tax Rates

Sales tax rates can apply to entire states, individual cities, individual counties, and even special tax districts. Cline and Neubig [CN99] estimate there are currently about 7500 different tax rates in the United States. All but four states impose a statewide sales tax. Most states permit local governments, e.g., cities, counties, and transit districts, to impose their own taxes. (See http://www.salestax.com/usmap.htm for a map showing which states charge sales tax and their different tax rates.) For example, consider California's sales tax structure. A statewide rate of 6.00% is supplemented by a 1.25% county and city tax rate. Other tax districts can further increase and complicate the tax rate. For example, all of Nevada County, California, participates in the 0.125% Nevada County Public Library District while only Truckee is in the Town of Truckee Road Maintenance District. Thus, the tax rate in the county is either 7.375% or 7.875%, depending on one's address. Washington State's similar tax system yields 331 different tax rates [Was99] with twelve different tax rates. Texas's five-level system of a base statewide tax, city taxes, county taxes, transit districts, and special district taxes (capped at a maximum 8.25%) is even more complicated because the districts need not obey county or city boundaries. For example,

> The De Leon Hospital District is in Comanche County, which has a county sales and use tax, but the district does not include all of Comanche County. The city of De Leon is totally within the De Leon Hospital District. The unincorporated areas known as Comyn, Downing, and Rucker are located within the De Leon Hospital District. The unincorporated areas of Comanche County in ZIP code 76444 is totally within the De Leon Hospital District. The unincorporated areas of Comanche County in ZIP codes 76442, 76445, and 76454 are partially within the De Leon Hospital District. Contact the Comanche County Tax Office at 254/893-2193 for additional boundary information [Tex99b].

2.2 Collection Costs

Unlike many other taxes, sales taxes collection costs are mainly borne by retailers, not government agencies. When selling to the public, companies calculate and collect sales taxes. The U.S. Supreme Court decisions of *National Bellas Hess v. Department of Revenue of the State of Illinois* (386 U.S. 753, 1967) and *Quill v. North Dakota* (504 U.S. 298, 1992) [U.S92] limit companies' responsibilities to collect a state's sales tax to those having a significant physical presence in that state. Thus, catalog and Internet-based companies need not collect sales taxes for most states. Companies having a physical presence in only one state face significant costs dealing with sales taxes. A Washington State Department of Revenue study showed the total cost of collecting and remitting sales tax for small retailers is 6.47% of collected taxes, for medium retailers 3.35%, and for large retailers 0.97% [Wel98]. Cline and Neubig estimate the cost of compliance for a large seller having a physical presence in fifteen states is 8.3% of collected taxes because of the complexity of the numerous tax regulations [CN99]. In addition to their obligations to collect sales taxes from their customers, many large business taxpayers purchase items without paying sales taxes but must file tax returns directly for these taxable purchases. Thus, states have legislated that businesses must bear the burden of the sales tax system. To compensate businesses for these costs, twenty-six states permit businesses to keep a portion of sales tax collections [Wel98, Chapter 8]. Cline and Neubig note these discounts frequently cover less than five percent of compliance costs.

3 Many Proposals for Simplifying the Sales Tax System

The increasing prevalence of ecommerce has raised awareness of the complexity of the sales tax system. The taxation of electronic goods, electronic delivery of goods, and anonymous transactions further complicate sales tax computations. States are also afraid of losing sales tax revenue to a quickly growing

Internet-based retail industry. The Internet Tax Freedom Law (Title XI of Public Law 105-277) forbid new taxes on Internet access, forbid taxes that would multiply tax electronic commerce, and established the Advisory Committee on Electronic Commerce to investigate the electronic commerce taxation issues. These electronic commerce tax investigations highlighted the current complexity of the sales tax system.

The Advisory Committee solicited proposals, including proposals regarding the taxation of transactions conducted through the Internet, asking to what extent the existing sales and use tax administration should be simplified. Numerous proposals have been submitted. Many of these recommended the creation of third-party software to calculate, compute, and collect sales taxes. We illustrate many of the issues by reviewing the relevant policy of the National Governors' Association.

The association recently adopted a policy calling for "a streamlined sales tax with simplified compliance requirements [that] will ensure that states are prepared to support the global electronic marketplace of the next century" [Nat99]. Among the provisions are

- one sales tax rate per state,
- uniform and consistent definitions of goods and services so that each state can decide which items may be taxed,
- simplification of the administration of sales taxes, and
- a requirement that remote sellers, i.e., sellers shipping items across state lines, collect sales taxes even when shipping across state lines.

The most relevant recommendation is

> to encourage the establishment of a system of independent third-party organizations that would be responsible for remitting taxes to states. Remote sellers would use a software package preapproved by the states that would calculate the tax due on the purchase based on the state rate where the item is sent, and electronically remit that tax to the collection organization [Nat99, Section 12.2].

Summaries of the key points of various other proposals appear in Table 1. For each proposal, we list

- the method used to determine the appropriate tax rate and tax jurisdiction,
- the technique used to determine an item's taxability,
- the proposed sales tax collection agency, and
- other issues.

Many proposals require one sales tax rate per state. These proposals, confusing the separate issues of tax rates and taxability of products, assume that having over 7500 different tax rates is a source of complexity that needs to be eliminated. The main source of complexity is the determination of which items should be taxed, which should not, and which are taxed at special rates. Most proposals propose creation of uniform product codes to be used to determine which products are tax-exempt. While creation of product codes could simplify sales tax computation, the author has not seen any proposal discussing the difficult task of determining an item's category. One expensive possibility is to require all manufacturers to relabel their products to include their categories. An alternative possibility to create the "zero-burden" systems of some proposals is that the trusted third-party that calculates sales taxes must be able to determine an item's category. Thus, these proposals ignore the issues of maintaining, updating, and using a very large database; given the very large volume of transactions, each database access must occur very quickly. If the

proposer	tax regions	taxability determination	collection agency	other issues
American Institute of Certified Public Accountants	state-based	uniform product categories	not mentioned	ignores determining item's category and collection issues
eCommerce Coalition	state-based	uniform product categories	not addressed	ignores determining item's category
Michael O. Levitt	not addressed	uniform product categories	"Trusted Third Party"	ignores determining item's category
Thomas A. McGuire's Zip Code Tax	zip-code-based	uniform product categories	"Zip Code Tax Agency"	ignores issue of determining item's category
Charles E. McLure, Jr., (Hoover Institution)	state and local-based	national product categories	either state tax agency or trusted third party	ignores determining item's category
MPP&W Consultants	ZIP+4	not addressed	by third party	ignores size of necessary database(s) and determining item's category
Taxware International	federal, state, local regions	uses catalog control numbers and "appropriate Commodity Codes"	Transaction Tax Server	illustration shows product tax rate matrix; no discussion of database sizes or existence of commodity codes

Table 1: Key Points of Some Proposals Submitted to the Advisory Committee on Electronic Commerce.

trusted third-party does not determine each item's category, the burden falls on the seller so the system requires much more than a zero burden. As with the NGA's proposals, most proposals posit the existence of one or two trusted third-parties to calculate, compute, and collect sales taxes. The author is not aware of any such proposal that addresses whether one or two companies should be given such monopolistic power. Some proposals posit computation and collection by state agencies or multistate agencies. The NGA proposal posits a "system of independent tax-party organizations," thus seeking to use market forces to reduce computation and collection costs.

4 Our Prototype

We will develop a prototype Internet-based software package calculating, but not collecting and remitting, sales tax due. There already exist numerous companies specializing in collection of moneys for Internet- and non-Internet-based businesses, e.g., CyberCash and Charge.com. The existence of off-the-shelf income tax programs and services, e.g., Quicken TurboTax and Kiplinger TaxCut, and electronic filing, e.g., Internal Revenue Service's Electronic Federal Tax Payment System and California Franchise Tax Board's's electronic services, demonstrate that electronic filing is technologically possible. For individual companies, Taxware International and Vertex provide software to compute sales taxes, rates, and jurisdictions, but these require extensive customizing and maintenance. Use of our prototype will be as simple as sending an address and a list of items to a WWW address and reading the response.

4.1 Expected Results

We predict our prototype will demonstrate which aspects of the National Governors' Association's policy are technologically possible to implement on today's engineering workstations. We predict that

- Requiring one sales tax rate per state is not necessary as long as tax jurisdiction boundaries do not cross ZIP+4 postal code boundaries. We note that it is the proliferation of tax jurisdictions each having its own tax rules, not the proliferation of tax rates, that causes difficulty when calculating sales taxes.

- Defining a uniform and consistent definition of goods and services is not by itself sufficient to permit fast, accurate computation of sales taxes. There also needs to be a mechanism to automatically determine to which category any particular item belongs. There are currently four barriers to implementing this mechanism:

  - The number of items available for sale is probably on the order of hundreds of millions of items.[1] Even if, on average, only ten percent of the states differ in their taxation of any particular item, the amount of information to store can easily exceed the amount of available virtual memory on today's 32-bit computers.
  - Many taxable items such as freight charges and electronically-delivered software are not currently labelled under any commonly-agreed system.
  - Even if every one of the hundreds of millions of potentially-taxable item is labelled, there still exists the difficulty of determining taxation of items that are sold only once, e.g., custom cuts of lunchmeat prepared at grocery store deli counters.
  - The number of items introduced and removed from the marketplace each year is probably on the order of ten million items [otC99, Table 896]. Even with a uniform definition of goods and services, each of these items must be categorized and the information distributed to the third-party organizations. Manufacturers and/or retailers would still bear these tax-compliance costs.

- Our technology will be an order of magnitude cheaper to use than existing technologies. Using the sales tax server will be as simple as formatting the input and output in the desired format; no other work on a company's part is necessary. Since we will be providing a service not yet commercially available, we do not expect to be able to quantitatively verify the claim.

4.2 Simplifications and Assumptions

Our prototype's purpose is to compute the sales tax to charge for a given list of items at the correct tax rate in a given tax jurisdiction. Our prototype, by definition, will not have data on all possible products and all possible tax rates and jurisdictions. This is because

- collecting data for any significant subset of salable items in even one tax jurisdiction would require hiring several professionals and several person-years of work,

- the current tax system is overly complex, and

- our goal is to discern and solve the critical issues in designing a program to calculate sales tax, not produce production-quality software.

[1] Inference based on 99.3 million new product introduction of consumer packaged goods from 1980 to 1997 [otC99, Table 896].

We will try to highlight

simplifications: to the current sales tax system necessary to automate tax calculation within a reasonable amount of programming effort and

assumptions: made to ease creation of our prototype.

4.3 Prototype from the User's Point of View

The prototype's purpose is to compute the sales tax to charge for any given list of items at the correct tax rate for any tax jurisdiction. Thus, the user, e.g., a retailer, must provide enough information to determine the tax rate and jurisdiction and also a list of possibly taxable items. The prototype server will return one of four possible answers:

- the sales tax due,
- an error indicating the input did not have the correct format,
- an error indicating the server was not able to compute the sales tax because sales tax rules are ambiguous or unknown items have been specified, or
- an error indicating an internal server error.

4.3.1 Input

Most input must be separated into clauses surrounded by matching left and right parentheses. Whitespace is not important except to separate terms. That is, input can be freely formatted using whatever newlines, tabs, and extra spaces the user desires.

An address can be specified as

```
(address (zip+4 78212-7200))
```

or

```
(address (street 715 Stadium Drive) (city San Antonio) (state Texas)).
```

The leftmost entry must be the keyword `address` and the rest of the clause should consist of enough subclauses to uniquely determine a ZIP+4 postal code. Each subclause consists of a keyword followed by information. Among the keywords are

`street`: specifies (possibly) a street number and a street

`city`: specifies a city's name

`state`: specifies a state either using the U.S. Postal Service's two-letter state abbreviation or using the state's entire name

`zip`: a five-digit ZIP postal code

`zip+4`: a nine-digit ZIP+4 postal code without a hyphen

`comment`: a comment to be ignored

Subclauses need not occur in any particular order, and keywords may occur any number of times but only the last occurrence of each keyword is remembered. Enough information must be provided for the United States Postal Services's ZIP+4 code lookup engine to determine the address's unique ZIP+4 code. If both a five- and nine-digit ZIP codes are specified, the nine-digit code is used. Capitalization is not important.

Items are specified in an `items` clause, which consists of zero or more `item` clauses. Each of the latter consists of enough subclauses to uniquely specify a particular item, the number of units purchased, and the price. Among permissible keywords for subclauses of `item` clauses are

`number`: specifies the nonnegative integral number of items purchased. The number may be specified using a decimal, octal, or hexadecimal representation. If this subclause is missing, the default value is one.

`item`: specifies the Universal Product Code (UPC) of the item(s). The code, represented as a string of decimal digits, is mandatory unless a `taxable` subclause is present.

`item-cost`: specifies the per-unit cost of the item.

`total-cost`: specifies the total cost of all items.

`taxable`: boolean, i.e., `true` or `false`, indicating whether the item is taxable. The sale of unique items makes this field necessary. For example, an original painting probably does not have a UPC code and the burden of determining taxability must be placed on the retailer.[2] If this subclause is absent, the server determines an item's taxability.

`comment`: a comment that is ignored.

Keywords may appear any number of times in an item clause, but only the last occurrence of each is remembered. Prices may be negative, e.g., indicating returning an item.

Assumption 1 *All prices will be represented as multiples of U.S. cents.*

To accommodate items that may be priced using fractions of cents, the `total-cost` keyword permits specifying the total cost of all items of the same type that are purchased.

Simplification 1 *We will not consider taxation of shipping, handling, and coupons.*

We have not researched current sales tax rules about these items so we cannot say whether significant simplification is necessary nor whether current guidelines would greatly complicate programming the server.

Input to the server consists of an `input` clause. In any order, it must have one `address` subclause and one `items` subclause and may have any number of `comment` subclauses.

4.3.2 Output

The server's output consists of an `output` clause. It must have one `tax` subcaluse xor one `error` subclause and may have any number of `comment` subclauses in no particular order. Example `output` clauses include

```
(output (tax 345))
```

and

```
(output (error unknown item) (comment (item 079068170000)))
```

Among the permissible keywords are

`tax`: specifies the amount of sales tax to charge, in U.S. cents.

`error`: specifies a reason why the sales tax cannot be charged. Example error messages include `bad input format`, `ambiguous tax laws`, `unknown item`, and `internal server error`.

`comment`: a comment to be ignored.

[2]If taxability is based upon product categories, this clause could easily be changed to relay the item's category. Since the number of product categories is likely to be much less than the number of UPC codes, converting categories to taxability status is a much easier task than converting UPC codes to taxability status. Thus, we omit this feature from our prototype. Regardless the retailer, not the trusted third-party, must determine the category.

4.4 Outline of the Server

The server's two major tasks are to determine the tax rate and jurisdiction and to determine the taxability of each item in the tax jurisdiction. We propose using ZIP+4 postal codes to determine tax rates and jurisdictions. Thus, the server consists of three major components:

- conversion of an address to its ZIP+4 postal code,
- converting a ZIP+4 postal code to a tax rate and tax jurisdiction, and
- determining the taxability of each item according to the tax jurisdiction rules.

We sketch each component below.

4.4.1 Tax Rates, Tax Jurisdictions, and ZIP+4 Postal Codes

As noted in Section 2.1, there are currently over 7500 different tax rates. The current sales tax system in some states, e.g., Texas, is too complicated to easily determine a tax rate. A simplification is necessary to easily automate determination of tax rates and jurisdictions.

Simplification 2 *Tax rates and jurisdictions may be uniquely determined using ZIP+4 postal codes.*

One could also use census tracts or any other politically-acceptable, fine-grained geographic measure that is easily determined by a software program. Among the advantages of using ZIP+4 postal codes are:

- Every U.S. address has already been assigned a ZIP+4 postal code.
- ZIP+4 postal codes are easily determined from addresses.
- ZIP+4 postal codes are very fine-grained; postal code regions typically consist of a city block or an office building. Thus, existing tax rates and jurisdictions can be very accurately approximated.

There are several practical difficulties with using ZIP+4 postal codes. Postal code regions regularly change.[3] An address does not necessarily correspond to the correct tax rate. A California tax board publication states

> It is not always possible to determine the correct rate based solely on the mailing address. For example, a customer may live near a county line and have a zip code and city name that routes mail to a post office in a neighboring county which may have a different tax rate. If you relied solely on the post office mailing address to determine the tax rate, you would assume the customer lived in a county other than the one in which he or she actually resides. As a result, you could apply an incorrect tax rate [Cal99, p. 1].

These "edge effect" difficulties will occur in any sales tax system with differing tax rates; consumers will seek to minimize taxes by shifting purchases to decrease tax rates. Thus, we do not seek to resolve this political difficulty.

We also assume only one address is needed.

Assumption 2 *Only one address is needed to determine the sales tax.*

Some tax jurisdictions charge taxes according to both a product's origin and destination addresses. For example, in Texas, local sales taxes are collected based on the business's location with the exception of transit taxes.

[3] Add more information!

> Transit tax is collected differently from other local taxes. The key element is where you deliver your service or product. Transit sales tax is due on all taxable items delivered from a place of business in a transit authority to a location in the same authority. Transit sales tax is not due on goods or services delivered to a location outside the transit authority. But you may need to collect transit use tax if that delivery is made to a location inside another transit authority. [Tex99a]

These tax rules could be incorporated into a production-quality server without significant additional space, time, or programming requirements.

4.4.2 Determining the ZIP+4 Postal Code

Given an address, the server will determine the corresponding tax rate and jurisdiction using the address's ZIP+4 postal code. For the prototype, we will use the U.S. Postal Service's (USPS) ZIP+4 Code Look-up Engine server if the ZIP+4 postal code is not already specified. USPS use guidelines permit only occasional use of the server, referring heavy users to companies that provide similar services. The production-quality server could use one of these services.

Addresses could be stored in `address` objects. The operations `address`es must support include:

- storing address fields, including the ability to change a field's contents. Most fields will probably be represented as `string`s or `unsigned int`s.
- print the address in a form understandable by the USPS ZIP+4 server, and
- yield the first three, five, and nine digits of the address's ZIP+4 postal code.

4.4.3 Determining the Tax Rate and Jurisdiction

Given a ZIP+4 postal code, the server determines the corresponding tax rate and jurisdiction. If this information was stored in a table with one row per postal code, this would just involve finding the code's row and returning the tax rate and jurisdiction. Unfortunately, the table's size requires too much memory to be practical.

The naïve implementation with one row per postal code could require up to one billion rows to store information about ten thousand tax rates, a redundancy factor of one hundred thousand. Using similarities between the geographic layout of postal codes and tax rates, we can reduce the table's size:

- Many states have one tax rate for the entire state, e.g., Kentucky, Michigan, and Indiana.
- ZIP codes having the same first three digits usually fall within the same state.
- Many tax regions contain many five-digit ZIP code regions.
- Few tax regions split five-digit ZIP code regions.
- Very few tax regions split five-digit ZIP code regions.

We will use three separate hash tables to convert from a ZIP+4 postal code to a tax rate and a jurisdiction. A hash table is a table permitting fast lookup. More precisely, each row consisting of a key and an item. Given the key, the hash table will yield the associated item. The hash table data structure uses randomness to quickly find (in constant time) the key and its associated item.

The first-tier hash table contains the nine-digit ZIP codes in different tax rate regions than the corresponding five-digit ZIP code. For example, suppose most of the 94301 postal code region is taxed at 8.25%

but the 94301-2301 postal code region is taxed at 4.56%. The nine-digit hash table will have a 943012301 entry of 4.56% while the five-digit hash table will have a 94301 entry of 8.25%. Associated with each nine-digit ZIP code is either a tax rate and jurisdiction or an indication of failure. If the lookup fails, we should use the second-tier hash table.

The second-tier hash table contains the five-digit ZIP codes with differing tax rates than their parent three-digit ZIP code tax rates. Associated with each five-digit ZIP code is either a tax rate and jurisdiction. If the lookup fails, we should use the third-tier hash table.

The third-tier hash table contains three-digit ZIP codes. Associated with each three-digit ZIP code is either a tax rate and jurisdiction. If the lookup fails, the server should indicate it could not compute the sales tax.

Three required data types are `ZIP`, `taxRate`, and `taxJurisdiction`. A `ZIP` is an opaque type stored in an `address` object. The `address` object should support

- storage of a ZIP+4 postal code,
- retrieval of the first three digits of the code,
- retrieval of the first five digits of the code,
- retrieval of the entire code (without the hyphen).

The hash tables need only consider `taxRate` and `taxJurisdiction` types as opaque types, i.e., no operations are necessary. Operations will be described below.

We omit discussion of storage of the ZIP code, tax rate, and tax jurisdiction information and creation of the hash tables.

4.4.4 Computing the Sales Tax on a List of Items

Given a tax rate, a tax jurisdiction, and a list of items, the most difficult step to compute sales tax is to determine the taxability of each item. A less important step is rounding the sales tax according to the jurisdiction's rules. We will first concentrate on determining taxability.

Simplification 3

Each item is either taxed or not taxed, depending only on the tax jurisdiction, not the tax rate.

Currently, sales tax rates vary according to both geography and item. For example, visitors to Harris County, Texas, pay a rental car sales tax rates of 15%, while most local residents pay a sales tax rate of 8.25% on most physical goods. Sales tax rates for gasoline, telecommunications, water, and other items also vary from the "default" 8.25% rate. As we shall see, determining an item's taxability already requires a huge amount of space even without having to record the specialized sales tax rate for individual items.

Simplification 4 *Every item is identified by a unique identifier.*

As we have discussed above, the trusted third party, not the retailer, determines the taxability of items and needs a way to uniquely identify any item sold by any retailer.

Assumption 3 *Every item has a universal product code (UPC) code.*

For this prototype, we assume that every item sold by a retailer has a UPC code. The Uniform Code Council administers twelve, thirteen, and soon fourteen digit numbers used to identify products. For example, most retail and grocery stores use laser scanners that automatically read UPC bar codes to compute each customer's purchase cost. Although decentralized, there already exist ways to distribute products' UPC codes from suppliers to distributors and retailers. It should not be difficult for trusted third parties to obtain similar information.

Conceptually, item taxability can be stored in a table having one row per item and as many columns as tax jurisdictions. Unfortunately, the table's size is prohibitively large. We conjecture that there are about one hundred million different items sold in the United States each year[4] and at least several hundred different tax jurisdictions so the table's size would be at least several gigabytes, larger than the largest possible virtual memory on many of today' 32-bit computers. A realistic upper bound is one billion items and ten thousand tax jurisdictions, requiring on the order of ten trillion bits of information. This estimate is based on a comment by Dennis Epley of the UC Council [UCC99]. He said most UCC members have less than one thousand items. Since the original UPC scheme had a six-digit company field, the upper bound of 10^9 follows.[5] This much information would not fit even in the storage devices of today's large engineering workstations.

We reduce the table's size by assuming that most tax jurisdictions will have the same taxation rules for a given item. That is, for any particular item, we store a default value of either taxed or not taxed and a list of tax jurisdictions with different values. At the very least the table's size will still be several hundred million bytes because it must store at least ten million items, each identified by at least a twelve-byte number and having a one-byte taxability status. In practice, we expect the table to be one to two orders of magnitude larger.

Because of the table's size, we will store the information as a set of files, rather than in memory or in one huge file. Each file identified by a six-digit number will contain taxability information of all items beginning with those six digits. This corresponds nicely with the UPC encoding of a company as its first six digits. Using a set of files permits easier addition and removal of entries and relatively quick search for individual items. A production-quality system might try to store the contents of most frequently used files in main memory and might also explore tradeoffs between the number of files and file sizes.

As for the address format, each item's taxability entry is surrounded by matching left and right parentheses. Whitespace is unimportant except to separate terms. For example, an item's taxability can be specified as

```
(taxability (upc 033200190103) (default taxable) (unknown nyc) (nontaxable ct wa tx ca))
```

The leftmost entry must be the UPC code and the rest of the clause should consist of enough subclauses to identify the item's taxability. Each subclause consists of a keyword followed by the information. Among the keywords are

`upc`: specifies the item's UPC code. This is a required subclause.

`default`: specifies the default taxability of an item using `taxable`, `nontaxable`, or `unknown`. If this subclause is omitted, an item is assumed taxable. If more than one clause is present, the value of the last one is used.

`taxable`: is followed by a (possibly empty) list of tax jurisdictions.

`nontaxable`: is followed by a (possibly empty) list of tax jurisdictions.

[4] See Section 4.1 for a justification of this number.

[5] Our claimed upper bound may actually be an underestimate for two reasons. The EAN-13 system uses thirteen digits, not the twelve of UPC codes. Secondly, the UCC is now introducing variable length company fields having six to eleven digits, presumably because it will otherwise run out of company prefixes [UCC99].

`unknown`: is followed by a (possibly empty) list of tax jurisdictions.

`comment`: a comment that is ignored

Excepting the `upc` subclause, subclauses need not occur in any particular order, and keywords may occur any number of times, but a tax jurisdiction may occur only once among all the subclauses. The algorithm for determining an item's taxability in a particular jurisdiction is:

1. If the tax jurisdiction explicitly occurs in a subclause, the keyword indicates the taxability.
2. Otherwise, if a default taxability is specified, that determines the taxability.
3. Otherwise, the item is assumed to be taxable.

If no entry for a particular item is found, we consider its taxability `notFound`.

We will implement a function that, given a tax jurisdiction and an item, returns whether the item is taxed, not taxed, is unknown whether taxed, or is an unknown item.

Given a function determining the taxability of any particular item, we compute sales tax on a list of items by summing together the total cost of all taxable items, multiplying by the sales tax rate, and rounding to the nearest cent.

Assumption 4 *All tax jurisdictions compute sales tax by first totalling the cost of all taxable items, multiplying by the sales tax rate, and then rounding.*

Although other algorithms are possible, we assume that all tax jurisdiction explicitly or implicitly require this algorithm. Even if this assumption is not true, we conjecture that the other algorithms have similar complexity and can easily be implemented in a production-quality server.

Few tax jurisdictions publish clear guidelines for rounding sales tax amounts. We conjecture that most jurisdiction require any amount greater than or equal to half a cent to be rounded up to one cent and any amount strictly less than half a cent to be rounded down. Some other plausible rounding schemes include

- rounding any fraction of a cent up,
- rounding any fraction of a cent down, and,
- if the fraction begins with a 5, rounding up one cent if the next digit is even and down otherwise.

According to a Board of Equalization employee, California's guidelines for rounding [oCSBoE99] are:

1. Compute the sales tax in cents.
2. Truncate the sales tax to have four digits after the decimal point.
3. Round up one cent if the amount after the decimal point is 0.5000 or greater.

To accommodate these rounding concerns, the `taxJurisdiction` object could have a private rounding function. Taking an amount representing cents, the function would return the rounded number of cents as an `Cost`. It could also have a member function returning the sales tax amount given a `taxRate` object and the cost of taxable items.

5 A Few Nontechnical Issues Concerning Trusted Third Parties

Even if all of our findings are implemented by legislatures and trusted third parties (TTP), retailers would still bear costs to deal with a sales tax system.

First, retailers' records would need to be audited to ensure they submitting all taxable transactions to a trusted third party for sales tax collection. This auditing burden is likely to be comparable to other auditing burden retailers already face.

Second, for trusted third parties to be as efficient as possible, these parties should be independent companies, not government agencies. These independent companies would compete to provide good service at reasonable prices with retailers paying for the services. Thus, the cost of complying with sales taxes would still fall on retailers, not the government. Even though governments would not have a direct incentive to reduce the cost of complying with sales tax laws, the service costs of trusted third parties, set by market forces, would clearly indicate these compliance costs. Thus, governments would be able to measure the tradeoff between increasing sales tax complexity to increase revenue and the resulting increase in revenue.

Third, a trusted third party must be trusted by its customers, contracting retailers. As part of its business, a trusted third party would have access to some or all of a retailer's sales information. Dishonest trusted third parties could sell this sales information to competitors or to suppliers. Concerned retailers could either contract with several trusted third parties so that no one TTP has access to all of its sales information or operate its own sales tax division. If sales tax computation and collection is performed by independent companies rather than government agencies, a market in tax computation and collection software could develop so concerned companies could have their own divisions.

Fourth, using trusted third parties raises legal liability issues. If a TTP computes the incorrect sales tax or fails to remit all collected sales tax, is the contracting retailer or the TTP responsible? Thus, legislatures must not only simplify the sales tax system but also establish a legal framework for TTPs.

References

[Cal99] California State Board of Equalization. California city and county sales and use tax rates. Technical Report 71, California State Board of Equalization, http://www.boe.ca.gov/-staxpubs.htm, April 1999.

[CN99] Robert J. Cline and Thomas S. Neubig. Masters of complexity and bearers of great burden: The sales tax system and compliance costs for multistate retailers. pdf file, Ernst & Young Economics Consulting and Quantitative Analysis, 08 September 1999. Prepared for the eCommerce Coalition.

[Nat99] National Governors' Association. Streamlining state sales tax systems. http://www.nga.org/-Pubs/Policies/EC/ec12.asp, Winter 1999.

[oCSBoE99] Employee of California State Board of Equalization. Personal communication, December 1999.

[otC99] Bureau of the Census. *Statistical Abstract of the United States*. U.S. Department of Commerce, Bureau of the Census, Washington, D.C., http://www.census.gov/prod/www/statistical-abstract-us.html edition, 1999.

[Tex99a] Texas Comptroller of Public Accounts. Guidelines for collecting local sales and use tax. http://www.window.state.tx.us/taxinfo/taxpubs/tx94_105.html, April 1999.

[Tex99b] Texas Comptroller of Public Accounts. Local sales and use tax. http://www.window.state.tx.us/taxinfo/local/spd.html, December 1999.

[UCC99] http://www.uc-council.org/news/ne_p111599.html, November 1999.

[U.S92] U.S. Supreme Court. Quill Corp. v. North Dakota. Available at http://supct.law.cornell.edu/-test/hermes/91-0194.Z0.html, 26 May 1992.

[Was99] Washington State Department of Revenue. Sales tax rates download data. http://dor.wa.gov/-gis/Add_Data/All_zips.exe, December 1999.

[Wel98] Mary Welsh. Retailers' cost of collecting and remitting sales tax. Technical report, Washington State Department of Revenue, http://dor.wa.gov/reports/retail/retailstudy.doc, December 1998.