Care in Classifying Computer Software Can Avoid or Reduce Property Taxes

Taxability may be based on whether software is an applications or operational program, or may depend on the software’s "tangibility."

NANCY S. RENDLEMAN AND CHARLES B. NEELY, JR.

In those states that impose property taxes on computer software, assessors and taxpayers face the difficult task of establishing whether particular types of software are assessable and, if so, the value of the software. The traditional cost, market-based, and income valuation methods frequently are inadequate to appraise software because of a lack of cost data, comparable sales, or income that can be allocated among various products and projects. Given the lack of uniform treatment by assessors, taxpayers may be able to take steps to reduce assessments of software subject to property taxation.¹

SOFTWARE CLASSIFICATIONS

Several classification systems have been used to determine whether computer software is subject to property tax. The primary legislative distinction is between "canned" and "custom" software. In some instances, taxation is based on whether the software is an "applications" or an "operational" program. Some statutes have merged the two standards to tax "custom applications" programs. Since many jurisdictions do not specifically address computer software in their property tax statutes, administrative bodies and the courts often have been left to fend for themselves. Some courts have relied on the "tangibility" of software as the touchstone to determine whether it is assessable. To further complicate the matter, both the legislative and the judicial solutions have been blurred by considerations that make the software distinctions more certain in theory than in practice.

The canned-custom continuum. The canned vs. custom distinction between software products turns on the source and development of the software. Canned programs generally are purchased from a retailer that sells the software as a mass-produced, pre-packaged, mass-marketed product—every customer receives an identical program. Custom software is designed and developed, either internally by the user or externally by another entity, to meet the particular needs of an individual user.

At least nine states rely on the distinction between canned and custom software to apply their property tax schedules, and most of these tax only canned software.² They apparently believe that custom software more closely resembles a tailored service than tangible personal property. The Rhode Island Supreme Court, for example, in ruling that custom computer software did not expose its owner to any property tax...
liability, stated that "a 'canned' ready-to-execute software program was tangible personal property," while the altered, custom software "involved an intangible-service element that the ready-to-execute...software did not."

One of the nine states has adopted the conflicting view under some circumstances. The New Mexico Taxation and Revenue Department has informally stated that canned software "is not considered personal property" and will not be taxed. Custom software, however, is subject to tax as personal property when owned by a business.4

Despite statutory and judicial reliance on the distinction between canned and custom software, the difference often is more theoretical than real. Canned software can be modified for the individual user’s needs, and the extent of the modification effectively creates a continuum ranging from pure canned to pure custom. Thus, practitioners may have to determine how substantially canned software has to be modified to be exempt as custom software. In some instances, the necessary alteration might be slight. In Washington, for example, canned software that has been modified by changes in coding may be considered "semi-custom," and exempt to the extent of the modification.5 The North Carolina Department of Revenue takes the position for sales and use tax purposes that where a customer merely requests advice from the seller concerning which of several canned programs best fit the customer’s needs, the software is a nontaxable custom program.6

A statutory definition of custom software could control the extent of the modification required to obtain an exemption. Of the states that rely on the canned-custom distinction, however, only Washington has defined custom computer software.7 In the remaining jurisdictions, the courts will be forced to resolve these issues with relatively little guidance from their state legislatures.

**Operational vs. applications.** While the canned-custom continuum often presents ill-defined distinctions, statutory schemes based on the function of the software offer a brighter line to distinguish among computer programs. Operational software (e.g., DOS) contains the basic instructions that tell a computer how to function. Such a program is essential to the productivity of the computer hardware itself. The interdependence of operational software and the hardware can make it difficult to distinguish between the two. On the other hand, applications software (the programs that solve particular problems by enabling computers to function as, e.g., word processors, data bases, or spread sheets) is easily identified. **WordPerfect** and **Lotus 1-2-3** are examples of applications software.

At least five states impose personal property tax on operational software, while providing an exemption for applications software.8 For example, California taxes only "basic operational programs."9 The California Board of Equalization has informally reported that the distinction between operational and applications software depends on the function of the program, and the value of software serving both functions should be apportioned.10

Absent a specific exemption for custom software, the imposition of property tax on operational software can result in taxation of such programs, whether canned or custom. In Maryland, both operational and canned software are subject to property tax, while only custom applications software is specifically excluded.11 Thus, lacking a similar exemption, custom operational software...

---

4 See also the authors’ earlier article on sales and use taxation of software: Rendleman and Neely, Jr., "States Seek to Broaden Sales and Use Taxation of Computer Software," 2 JMT 196 (Nov/Dec 1992).


6 Washington Code Ann. §84.36.600 (West 1994).

7 43 N.C. Admin. Code §7B.0116 (repealed, effective 10/1/93, but still followed by the Department).

8 Washington Code Ann. §84.36.600 (West 1994). Washington excepts "embedded software" (i.e., software that may not be removed without terminating the operation of the system) from the exemption for custom software.

9 California, Indiana, Kansas, Nebraska, and Tennessee have adopted this approach. See "Personal Property Taxation of Software," supra note 2.

10 Telephone conference with California Bd. of Equalization personnel (10/5/94).


12 See also the authors’ earlier article on sales and use taxation of software: Rendleman and Neely, Jr., "States Seek to Broaden Sales and Use Taxation of Computer Software," 2 JMT 196 (Nov/Dec 1992).
ware would be taxable in Maryland. Several state courts have examined the operational and applications functions of software to rule on its taxability. In *Compuserve, Inc. v. Lindley*, an Ohio appellate court found that operational software programs “have a value that is to be considered an essential portion of the computer hardware and are therefore taxable as tangible personal property in conjunction with the hardware.” The court noted that the operational software was inextricably linked to tangible hardware, despite its own intangibility. Thus, the court followed a line of Ohio cases that had held that “intangible incidental costs which enhance the value of the tangible personal property are considered part of the true value of the business personal property.”

The Kansas Supreme Court employed a similar rationale in *In re Protest of Strayer.* The court concluded that operational software was taxable under the Kansas property tax scheme “in conjunction” with the hardware. According to the court, however, applications software was not tangible property and therefore was exempt from taxation.

Maryland also has relied on the operational-applications distinction. In *Greyhound Computer Corp. v. State Department of Assessments*, Maryland’s highest court ruled that, unlike operational software, the value of applications software could be distinguished from the value of computer hardware. Only the value of “tangible computer software” linked with the computer hardware was subject to property taxation.

The tangibility touchstone. Most states either entirely exclude intangibles from the property tax base or tax only particular types of intangible property, such as bank deposits. Accordingly, if the courts in such a jurisdiction find software to be an intangible, it will be exempt from taxation. Absent legislative guidance, however, determining whether software is tangible or intangible can present problems.

Although assessors have had some success clearing the intangibility hurdle by linking operational software to the hardware on which it was installed, courts usually have held computer software—particularly custom and applications software—to be intangible property exempt from personal property taxation. In doing so, the courts have applied several familiar tests, primarily in the sales and use tax context. That reasoning generally should apply in the property tax arena as well. As case law develops in this area, however, these tests of tangibility—particularly the mode-of-transmission test—may not apply to a property tax analysis.

Buying knowledge. The knowledge-rationale and essence-of-the-transaction tests emphasize the information acquired rather than the means of acquisition. The knowledge-rationale test stands for the proposition that computer software is merely a means to transfer information from the creator of the data to the end user. For example, in *Commerce Union Bank v. Tidwell*, the Tennessee Supreme Court held that the purchaser of software bought the knowledge stored on the magnetic tapes, punch cards, and diskettes, rather than the tapes, cards, and diskettes themselves. The test presumes that once the information has been installed on the computer, the tapes, cards, or diskettes used to transmit the data can be destroyed. Thus, the purchaser of software values the information on the tangible medium, not the tangible medium itself, and, as an intangible, the information is not subject to property tax.

The essence-of-the-transaction test focuses on the intent of the end user, not the medium that the end user acquires. In *First National Bank of Fort Worth v. Bullock*, for example, the court held that the

---

13. The Ohio court also cited Rev. Proc. 69-21, 1969-2 CB 303, which is interpreted as setting forth the IRS’s position that software is an intangible asset. See note 43, infra, and accompanying text.


15. 320 A.2d 52 (Md., 1974).


17. Under recently enacted Ga. Code Ann. §48-1-1 (West 1994), computer software is tangible personal property. Nevertheless, the Georgia statute taxes software only to the extent of the value of the medium on which the program is stored or transmitted. Thus, generally the taxable value is minimal compared with the value of the software itself.


19. See, e.g., *District of Columbia v. Universal Computer Associates, Inc.*, 465 F.2d 615 (CA-D.C., 1972) (50% of the value of the software/hardware package attributed to software, which was not taxable); *Honeywell Information Systems, Inc. v. Maricopa County*, 575 P.2d 801 (Ariz., 1978) (computer software was intangible property not includable in the value of the tangible computer equipment); *Greyhound Computer Corp. v. State Dept. of Assessments*, supra note 15 (applications software was separable from value of hardware); *Computer Associates Int’l, Inc. v. City of East Providence*, supra note 2 (modified and custom programs involved intangible-service element and were not subject to personal property taxation).

20. For a more complete discussion of the tests in the sales and use tax context, see Rendleman and Neely, Jr., supra note 1.

21. 538 S.W.2d 405 (Tenn., 1976) (sales tax).

buyer intended to purchase instructions for performing deposit, lending, and general accounting functions, rather than a tangible product. Thus, the software was not taxable.

The knowledge-rationale and essence-of-the-transaction tests do not depend on the classification of the software involved. Under either test, the program may be canned or custom, operational or applications, and still be intangible property exempt from taxation. For example, in Northeast Datacom, Inc. v. City of Wallingford, the Connecticut Supreme Court ruled that canned and internally developed custom software were not tangible property similar to books, and therefore were not subject to property tax. The court noted that "the fact that tangible property is used to store or transmit the software's instructions does not change the character of what is fundamentally a classic form of intellectual property." The Connecticut court's approach is entirely consistent with both the knowledge-rationale and essence-of-the-transaction tests.

Mode-of-transmission test. The mode-of-transmission test takes into consideration the tangible nature of magnetic tapes, punch cards, and diskettes used to store or transmit the software, as well as any accompanying instruction manuals. In the vast majority of sales and use tax cases, the nature of the mode of transmission had little effect on the judicial interpretation of whether software was tangible. Nevertheless, courts in a minority of states have emphasized the mode of transmission. In Chittenden Trust Co. v. King, the Vermont Supreme Court held that regardless of how the software could have been transferred, the way in which it was transferred was controlling. Thus, under the mode-of-transmission test, software transferred electronically (e.g., by modem) might escape property taxation, while programs on diskettes will be assessed. Still, the mode-of-transmission test examines the nature of the property at transmission, rather than at assessment and valuation, so its applicability to property taxation may be limited.

Custom software as a service. When computer software is compared with other products such as books and movies, sales and use tax arguments apply with equal force in the property tax context. It has been said, for example, that custom software is more like a will drafted by an attorney than a book written for general distribution because the computer program addresses the needs of an individual user. Thus, to preclude findings of tangibility and taxability, a taxpayer might argue that custom software represents a service. While this position appears to concede that canned software is an appropriate object of property taxation, the concession would cost the taxpayer nothing in those states that specifically subject canned software to property taxation.

North Carolina's view. North Carolina's approach to the valuation and taxation of computer software defies traditional classification. Generally all property, whether tangible or intangible, is subject to property taxation. As taxing authorities in the state became more aggressive in assessing software, however, and as the difficulties inherent therein became more evident, an industry coalition lobbied the North Carolina General Assembly to provide more favorable treatment. With certain exceptions, the state now generally excludes computer software from property taxation. North Carolina continues to tax "embedded software" that is "not intended for removal without terminating the operation of the system," as well as software purchased from an "unrelated" person and capitalized on the taxpayer's books. A person is not related to the purchaser if the two are neither commonly owned nor own an interest in each other. Thus, if capitalized, both canned and externally developed custom software face property taxation, while internally developed, non-embedded custom software is not taxed. The exclusion of internally developed custom software avoids difficult valuation problems (discussed below).

While the North Carolina classification scheme does not fit within the traditional framework, it may indicate that as states recognize the difficulties in assessing computer software—particularly custom or internally developed software—their legislatures may attempt to tax software by drawing from the theories and policies of each of the traditional classifications, rather than using a single classification scheme, or may choose to exempt those types of software most difficult to assess fairly.

17 Id. at §§105-275(40)(a) and (b).
Market-based method. The market-based method relies on a comparison of recently transferred property of a substantially similar nature to the property being valued.\textsuperscript{31} Comparable sales in the market generally offer the most accurate valuation. Nevertheless, the market-based method is not without its difficulties. The method applies only where the appraiser can find sales of similar software or prices of similar software in vendors’ catalogs. Canned software marketed in various outlets presents an obvious opportunity to use the market-based method. Where the canned software has been modified, however, or the software is custom designed, the market-based method breaks down. By definition, custom software is designed for only one user whose needs might be unlike those of anyone else in the market.

Several states have recognized the difficulty of using the market-based approach. For example, the Maine Department of Revenue values internally developed custom software at cost, and uses the market-based method for other types of software. Absent comparable sales, however, Maine relies on other valuation approaches, including the cost method.\textsuperscript{32} Although the market-based method may offer greater accuracy, it may not apply to a substantial percentage of available software.

Income capitalization method. The income capitalization method values the stream of income derived from property. The annual income for the remaining life of the property is estimated and a capitalization rate is applied to calculate the market value.\textsuperscript{33} Consequently, the income capitalization method is best suited for properties that generate consistent cash flows.

While computer software can be a productive asset that contributes to the net income of a business, it may be exceedingly difficult to allocate that income among various assets. Other than the developer of the master copy of a canned program, few owners of software would have an income stream susceptible to accurate allocation. In any event, the rapid obsolescence of computer software might defeat any effort to estimate future income. These substantial difficulties have limited the application of the income capitalization method with regard to software.\textsuperscript{34} Washington, which might have generated significant tax revenue by using the income capitalization method to value the property of Washington-based software giant Microsoft Corp., exempted master copies of software from property taxation.\textsuperscript{35}

Cost method. Under the cost method, the appraiser computes the reproduction costs of the property, adjusted for depreciation.\textsuperscript{36} The cost method is frequently relied on to value software as well as other personal property, even though functional, technological, and economic obsolescence may not be adequately factored into the appraisal.\textsuperscript{37}

The states take a variety of approaches in applying the cost method. For example:

- The Ohio Department of Taxation recommends that canned and operational software be assessed at 100% of the purchase price in the first year, and depreciated over periods ranging from six to 14.8 years to a residual value of 16.8% to 20% of

\textsuperscript{31} Telephone conference with Maine Dept. of Revenue personnel (8/24/94).

\textsuperscript{32}\textit{Dimensions of Value, supra} note 31.

\textsuperscript{33} California attempts to allocate income to the software by multiplying the taxpayer’s capitalized earnings by the ratio of taxable software (valued using the cost method) to aggregate property owned (valued using various methods depending on the particular item of property). Telephone conference with California Bd. of Equalization personnel (10/3/94).

\textsuperscript{34} Wash. Code Ann. §84.36.600(4) (West 1994).

\textsuperscript{35}\textit{Dimensions of Value, supra} note 31, at pages 22-23.

\textsuperscript{36} Id., at pages 13-14, 23.
cost (depending on the software’s use).  

- The Texas State Comptroller’s Office uses the cost method to avoid the difficulties inherent in the income and market-based methods. Local appraisers use depreciation schedules based on three- to five-year lives and 10% to 20% residual values.  

- Although Florida does not have a state-wide personal property tax, local officials generally rely on an average measurable life of five years and a residual value of 20% for computer software.  

- The North Carolina Department of Revenue recommends that software be depreciated (effective this year, at an accelerated rate) over five years to a 15% residual value.

Despite the diverse approaches used by assessors in applying the cost method, few reported software valuation cases have focused on the mechanics involved. In *Burlington Northern R.R. Co. v. Bair*, the railroad challenged Iowa’s tax assessment on its property, including software. The court held that software was an intangible, and thus not taxable under Iowa’s property tax system. The court then had to determine the value of the software to deduct it from the assessed value of Burlington’s taxable property. Without discussion, the court stated that “[i]t is acceptable to use the cost approach to arrive at values for the computer software.” The *Burlington Northern* decision thus offers little insight into the mechanics of the cost method as it applies to software.

In *District of Columbia v. Universal Computer Associates, Inc.*, the court faced a similar issue. The taxpayer had purchased a data-processing unit and sought to reduce the assessed value of the property by deducting the value of the computer software. The court ruled that the software was an intangible not subject to property taxation, and was left with the valuation issue. According to the court, the value of the software “represented the cost of services rendered by IBM [the external developer] in the development of the tax program package.” Applying the cost approach, the court deducted the software development costs from the assessed value of the unit. The court, however, offered no significant guidance on valuing the software, since it merely ruled that, given the evidence, it was not unreasonable for the lower court to have found that the value of the software was 50% of the cost of the hardware/software package.

While canned software and externally developed custom software pose valuation challenges even under the cost method, internally developed software presents even greater difficulties. Although most assessing jurisdictions use a depreciated-cost approach to valuation, few taxpayers can provide meaningful cost data for their internally developed software. The taxpayer is rarely able to document the man-hours expended and the dollars invested in the production of software. The problem is exacerbated by the inescapable realization that some significant costs may be of little utility to the ultimate product, while relatively minor costs might be incurred in developing revolutionary software.

In *Unimet Corp.*, the court examined the aggregate number of reusable lines of computer code. Although experts offered conflicting studies, the parties and the court apparently agreed that only the reusable lines had any value. The court then deducted depreciation attributable to functional obsolescence. Finally, to calculate the value of the software, the court multiplied the depreciated number of reusable program lines by an average cost per line, which also had been widely disputed by two experts who had extensively studied the production rates and costs for program lines. The court’s reproduction cost analysis might be attacked for its assumption that value is correlated with the number of lines produced or effort expended. Nevertheless, it represents an instructive attempt to apply the cost method in the absence of sufficient information.

Some academics have attempted to make things even more complex. The cost method presented in *Unimet* is in some respects a simplified version of the

---

38 Telephone conference with personnel of the Ohio Dept. of Tax’n, Property Tax Division (9/26/94).

39 Telephone conference with personnel of the Texas State Comptroller’s Office, Property Tax Division (9/26/94).

40 Telephone conference with personnel of the Florida Dept. of Revenue, Tangible Personal Property Section (10/3/94).

41 815 F. Supp. 1223 (DC Iowa, 1993).

42 Note 19, supra.

43 In Rev. Proc. 69-21, supra note 13, the IRS noted that “[t]he costs of developing software ... so closely resemble the kind of research and experimental expenditures that fall within the purview of section 174 ... as to warrant accounting treatment similar to that accorded such costs under that section.” Thus, IRS recognized that a taxpayer may either expense all costs attributable to the development of software or amortize the expenditures over at least five years or any determinable useful life. While the Service’s policy does not directly deal with the valuation of software, it recognizes that software development is experimental and may well incur costs that have no bearing on the value of final product.
constructive cost model (COCOMO), an algorithmic software cost estimation model advanced by one authority. COCOMO attempts to calculate the man-months required and to estimate the dollar cost to produce the software, and then uses an effort equation for different types of software to arrive at a final calculated value. COCOMO, however, would be an expensive, complicated, and impractical valuation method for any tax assessor to use on a routine basis. In addition, COCOMO has been criticized for not rewarding efficiency or recognizing the time-consuming necessity of reprogramming and correcting existing lines of code. Thus, where courts or assessors adopt the cost method to estimate the value of software, they probably are more likely to use the simpler, but still complex, Unimet standards.

Even using the costly Unimet methodology, the value of internally developed software remains uncertain. The value may be calculated while software is under development, at the moment of completion, on its first use by the developer, or on distribution to third parties. Internally developed software also undergoes modification and rapidly becomes obsolete along with every other type of software. Thus, the value of software as calculated under the cost method might differ greatly depending on when the valuation was completed.

PLANNING
Steps that can be taken to minimize or avoid a property tax liability on computer software include the following:

1. Purchasers or licensees of computer software and hardware should ensure that these components are separately invoiced, or at least individually itemized on the bill. Similarly, all software should be separately identified to distinguish between custom and canned programs.
2. Whenever possible, software should be delivered to the buyer through tangible media, such as over telephone lines or by direct keyboard entry, rather than on diskettes or tapes. This could aid in the characterization of the software as a service, or at least as an intangible.
3. The sales documents for custom software should clearly identify the services provided by the seller in designing the program for the buyer’s particular use.
4. Any charges for modifications to canned software should be billed separately as for services or custom software.
5. Where appropriate, a portion of the purchase price might be allocated to a maintenance contract.
6. The local definition (statutory or regulatory) of custom software should be reviewed, along with related case law. The modifications necessary to transform canned software into a custom program may be relatively minor. In a few instances, a modification simply to adapt the software to the particular user’s computer hardware might suffice to render the software exempt.

CONCLUSION
The procedures for classifying and valuing software are complex, time-consuming, and expensive. Perhaps the significant complexity and expense will force both taxing jurisdictions and assessors to ease up, particularly with respect to internally and externally developed custom software. Clearly, intense competition and rapid evolution of software, as well as other changes in the software industry, indicate that states should carefully review their taxation and depreciation tables to ensure that the lives assigned to software are not excessive and residual values are not too high.

MARYLAND
Courts Set Standard for Manufacturing Exemption

Two Maryland holdings (a recent Tax Court case and the earlier Court of Appeals decision on which it relied) have considered personal property manufacturing exemptions for assets used in the film and video industry. These cases are significant for all industries, however, in evaluating the standards to be used in determining whether a particular operation is “manufacturing.” In both cases, the courts made it clear that qualification is based primarily on whether a substantial transformation has taken place in form and use from the product’s original state. Only if the substantial transformation test does not resolve the matter are other, secondary factors considered (i.e.,

---

46 74 Bkprtcy. Rptr. 156 (Bkprtcy. DC, Ohio, 1987).
50 Ohio calculates the value of internally developed custom software by examining the man-hours and dollars invested to produce the software. Telephone conference with personnel of the Ohio Dept. of Tax’n, Property Tax Division (9/26/94).