



EMBEDDED SOFTWARE TRENDS

SOFTWARE TAXATION DEVELOPMENTS

Maximizing Favorable Tax Treatment for Software Embedded in Purchased Assets

The tax treatment of software, on both the federal and state levels, has not kept pace with the accelerating use of software to control machinery and equipment.

About This Article

The article appears in the August 2006 issue of *The Journal of Taxation* magazine, published by Thomson Tax & Accounting.

Pellegrino & Associates LLC specializes in the valuation of software embedded in high-tech equipment for state and federal tax purposes.

If you would like to receive hard copies of this or other items that we publish, or if you are not on our mailing list and would like to receive future publications, please call us at (317) 566-8199 or email us:

info@pellegrinoandassociates.com.

Pellegrino & Associates, LLC
10333 North Meridian Street
Suite 150
Indianapolis, IN 46290

Voice: 317.566.8199
Fax: 317.569.7144

www.pellegrinoandassociates.com

Copyright © 2005
Pellegrino & Associates, LLC
All rights reserved.

Software is embedded in everything today, enabling the functionality of products that touch every industry. Drive a car and software embedded in a half-dozen computers installed in the car control components from the engine to airbag systems. Software guides farming equipment, automatically controlling soil testing. It dispenses seed and fertilizer, and even can use GPS coordinates to steer as a tractor traverses a field. Software takes raw ultrasound data and turns it into a meaningful image of your next child. The pervasiveness of software will only grow, fueled by a continued proliferation of general-purpose microprocessors and requisite embedded software.

Software use in product development has changed the way vendors implement product features. Before software, vendors built product features discretely using hardwired electronic or mechanical devices. Before software-controlled fuel-injection systems, cars had feedback carburetors that used dozens of vacuum hoses and mechanical systems to create the optimal air-fuel ratio under various operating conditions. Today, software controls this as well as many other new vehicle features. For example, the only difference between a 300 HP engine and a 350 HP engine may be an internal software switch. All physical components of the engine are otherwise identical. The enabling power of software applies to other equipment, such as medical imaging systems, casino equipment, manufacturing equipment and robotics, retail point-of-sale systems, and even telephones. All have substantial amounts of embedded software controlling their core functionality.

This embedded software generally has favorable tax treatment at both the state and federal level—to the extent that a taxpayer can establish the software's value. This favorable tax treatment comes in two forms.¹ At the state level, software—which generally is considered intangible property—may be exempt from tangible personal property taxes. At the federal level, companies can benefit from faster expensing of the capitalized software's value, which is amortized over 36 months instead of depreciated over five or seven years (the typical asset life of the equipment in which it is embedded). All the taxpayer needs to know to capture these benefits is the embedded software's value.

¹ A third possible tax benefit may be through a reduction in sales and use taxes. An in-depth discussion of this topic is beyond the scope of this article.

The problem is that most vendors do not itemize the cost of embedded software on their invoices when they bill their clients. There are many reasons for this, including customers' unwillingness to pay for software and the vendor's competitive concerns.² Consequently, it is difficult for a taxpayer to establish the value of the embedded software and then use that valuation for tax purposes.

Example: A taxpayer buys a new Dell computer for \$800 and also chooses to purchase Microsoft Office in the same transaction. Microsoft Office is clearly software subject to favorable tax treatment. The taxpayer pays about \$200 for Microsoft Office above the cost of the computer hardware. Dell, however, lumps all hardware and software costs together when it invoices the taxpayer for \$1,000. Because Dell does not state the cost of Microsoft Office on the taxpayer's invoice separately from the hardware, the taxpayer can lose the tax benefits of the software. This would not occur if the taxpayer bought the items separately.

Transactional structure can therefore become an important part of recognizing software's tax advantages. This is significant since tax jurisdictions routinely elevate transactional substance over form. Because transactional structure can be important, traditional ways of viewing capital equipment do not apply, particularly with respect to tax matters.³ Nor do traditional ways of viewing computer software, at least insofar as simple, easy-to-understand application programs such as a word processor or accounting system. Such a myopic view can in fact be costly to a taxpayer. Software takes many forms and performs many functions outside of what one typically witnesses in an office environment with an accounting system or word processor. It is common for embedded software performing the functions demonstrated in the opening examples to comprise 30%, 40%, even 50% or more of the value of a piece of modern capital equipment.

Recognizing the tax advantages of this software can improve the equipment's return on investment and create value—in some instances, up to 10% or more of the equipment's acquisition cost. The problem is that establishing the software's value to capture this beneficial tax treatment has historically been difficult. A basic misunderstanding of software drives this issue, followed by a historical lack of valuation methods for embedded software, inconsistent regulations, and conflicting court cases that create confusion at the federal and state level.

The Software Misunderstanding

Software is nothing more than a high-level abstraction that humans use to represent a predetermined way to transfer voltages in a microprocessor. This high-level abstraction makes it easy for us to use and program computers. Nevertheless, the reality is that software represents nothing more than human knowledge.⁴

Embedded software generally has favorable tax treatment at both the state and federal level—to the extent that a taxpayer can establish the software's value.

² See Banham, "Surprise Inside," CFO Magazine, 10/1/05, available at www.cfo.com/article.cfm/4444459/c_4448927.

³ See Santosus, "Cut Your Tax Bill When You Buy IT," CIO Magazine, 6/1/05, available at www.cio.com/archive/060105/tl_spending.html.

⁴ See Kuo, "Sales/Use Taxation of Software: An Issue of Tangibility," High Technology L.J. (U.C. Berkeley, Boalt Hall School of Law, 1987), page 3.

In fact, early tax cases made this distinction using the "knowledge test."⁵

Because software represents human knowledge, it is therefore intangible, being something that you cannot touch, see, or feel directly. Further, software does not have any physical form of its own. There are physical manifestations of software on things like hard drives, and printouts of source code, but these physical manifestations are incidental to the inherent value of software. Further, some software may be worthless or very valuable depending on physical form. An old computer tape with an important product algorithm is worthless if nobody has hardware that can read the tape. A printout of a software algorithm may be very valuable, especially if it describes something central to a patent claim.

Ultimately, people do not purchase software for its physical manifestation but rather for the knowledge it represents. Further, physical manifestation, which became an issue in tax cases in the 1980s under the "container test,"⁶ is an outdated and erroneous concept, given that much software today is sold in electronic download-only form or through "load and leaves," where there is never a physical manifestation of the software acquired by the buyer. In fact, certain courts recognized that taxpayers who download software instead of purchasing it on a tangible medium could receive favorable tax benefits.⁷ As noted above, tax jurisprudence routinely considers the substance of a transaction over its form; to ignore the substance of a software acquisition transaction in favor of a form test via some physical container appears both arbitrary and inconsistent.

Taxpayers share the blame in the controversy. Through the years, taxpayers tested the limits of the tax treatment of software in various ways. A lot of the controversy focused on the tangible vs. intangible treatment of software. Taxpayers advocated for both sides, depending on which argument provided them with the more favorable financial outcome.⁸ For example, taxpayers argued that software was tangible to qualify for investment tax credits.⁹ In the next breath, taxpayers argued that software was intangible and excludable from sales and use and personal property taxes.¹⁰

Thus, software treatment has a checkered judicial past at both the state and federal level. One commentator notes "the judiciary is caught in the middle of the fray, burdened with the responsibility of interpreting the different types of software.... [L]acking guidance from precedent or legislature, the courts have further complicated the area by drawing inappropriate analogies and misapplying various tests."¹¹

The problem is that most vendors do not itemize the cost of embedded software on their invoices when they bill their clients

⁵ See *District of Columbia v. Universal Computer Associates, Inc.*, 465 F2d 615 (CA-D.C., 1972).

⁶ See *Kuo*, *supra* note 4, page 6; *Comptroller of the Treasury v. Equitable Trust Co.*, 464 A2d 248 (Maryland, 1983); *Chittenden Trust Co. v. Commissioner*, 465 A2d 1100 (Vermont, 1983).

⁷ See *Equitable Trust Co.*, *supra* note 6.

⁸ See Pellegrino, "Embedded Software Valuation," 9 *Valuation Strategies* No. 2 (November/December 2005), page 12, at 15.

⁹ See *Texas Instruments Inc.*, 39 AFTR 2d 77-1383, 551 F2d 599, 1 EBC 1752 (CA-5, 1977).

¹⁰ See *Universal Computer Associates, Inc.*, *supra* note 5; *Equitable Trust Co.*, *supra* note 6.

¹¹ See *Kuo*, *supra* note 4.

Tax jurisdictions did not help since they generally looked to treat software in a way that best filled their coffers.¹²

Standards for the valuation of software have opened a new opportunity for taxpayers to claim the tax advantages of embedded software using formal, independent third-party appraisals for software value when an equipment vendor cannot or will not state it. Unfortunately, the IRS and many tax jurisdictions rely on outdated language despite changes in the software market, changes to valuation practices, and changes to valuation methodology.

Internally Developed Software

Tax jurisdictions have historically made a distinction between purchased/leased software and software developed internally by a taxpayer. The taxation issue is equally problematic for both, although the problems arise in different ways. Tax issues relating to embedded software do not apply in the case of self-developed software, because the taxpayer generally purchases embedded software. In both instances, the tax issues create inconsistencies whether one is reporting results under generally accepted accounting principles (GAAP) or for tax purposes. Statement of Financial Accounting Standards (SFAS) No. 86 covers the accounting treatment for the costs of internally developed computer software to be sold, leased, or otherwise marketed by a company under GAAP (i.e., internally developed software):

All costs incurred to establish the technological feasibility of a computer software product to be sold, leased, or otherwise marketed are research and development costs. Those costs shall be charged to expense when incurred as required by FASB Statement No. 2, Accounting for Research and Development Costs.... For purposes of this Statement, the technological feasibility of a computer software product is established when the enterprise has completed all planning, designing, coding, and testing activities that are necessary to establish that the product can be produced to meet its design specifications including functions, features, and technical performance requirements.¹³

SFAS No. 86, published in 1985, does not reflect the reality of the software development industry accurately. The dissenting opinion by Donald Kirk and David Mosso, issued at the time FASB published the standard, provides arguments as to why the standard does not reflect reality. Both Kirk and Mosso made rather remarkable observations in their dissent:

- SFAS No. 86 unduly restricts the capitalization of software costs.

Taxpayers share the blame in the controversy over the tax treatment of software by focusing on tangible vs. intangible treatment

¹² *Id.*

¹³ See Statement of Financial Accounting Standards (SFAS) No. 86, "Accounting for the Costs of Software to be Sold, Leased, or Otherwise Marketed" (FASB 1985), page 5.

- SFAS No. 86 extends the research and development classification of SFAS No. 2 ("Accounting for Research and Development Costs") to a major class of routine production activities.
- SFAS No. 86 permits significantly different amounts of capitalization depending on a company's choice of production methods.
- SFAS No. 86 results in the expensing of most computer software costs, even though software is a significant, and often the only, revenue-generating asset of many companies. Assessing the probability of future benefits from computer software is difficult in the software industry, but no more difficult than in some tangible output industries such as fashion clothing and oil and gas drilling, or even in other creative process industries such as motion pictures. In each of these, capitalization of costs is accepted despite the inherent uncertainties.¹⁴

Kirk and Mosso were correct in their predictions of how companies would respond, claiming a disproportionately large amount of R&D activity

Kirk and Mosso were correct in their predictions of how companies would respond, claiming a disproportionately large amount of R&D activity. Consider the situation of Microsoft's software accounting for GAAP and tax purposes.

Microsoft is a good case study since it is the largest software company in the world and expends a significant portion of its resources on internally developed software for sale.¹⁵ As a matter of routine, Microsoft must distinguish between the software development costs it must expense and those it must capitalize. According to Microsoft's 2005 Annual Report, it spent \$6.60 billion, \$7.78 billion, and \$6.18 billion on R&D during fiscal years 2003, 2004, and 2005, respectively.¹⁶ This R&D activity includes costs associated with payroll, employee benefits, stock-based compensation, and other headcount-related costs associated with product development, third-party development and programming costs, localization costs incurred to translate software for international markets, and the amortization of purchased software code and services content. Microsoft's position is essentially that all software development activity related to every software product that Microsoft sells in the market is R&D.

Windows XP is the product of interest for this analysis. Microsoft has been selling Windows XP in the market for five years and spent about \$1 billion in development costs.¹⁷ Microsoft expects to sell Windows XP until its next generation operating system (Windows Vista) is available in mid-2007.¹⁸ Given this fact, one would expect to see some asset on Microsoft's balance sheet to reflect the Windows XP

¹⁴ *Id.*, pages 8-9.

¹⁵ Though Microsoft is used as an example for this article, a sampling of public filings by the largest software companies in the world show that they each handle their treatment in substantially the same way as is described here.

¹⁶ The Microsoft Corporation 2005 Annual Report, page 12, available at www.microsoft.com/msft/ar.mspx.

¹⁷ See Swift, Bershad, and Levy, "Improving the Reliability of Commodity Operating Systems," Proceedings of the 19th ACM Symposium on Operating Systems Principles (Bolton Landing, N.Y., 2003), page 10.

¹⁸ Microsoft 2005 Annual Report, *supra* note 16, page 9.

asset, net of five years' worth of ratable amortization over some discrete useful life. Instead, per GAAP under SFAS Nos. 2 and 86, Microsoft expenses all costs, as seen in its annual report:

"We account for research and development costs in accordance with several accounting pronouncements, including SFAS No. 2, Accounting for Research and Development Costs, and SFAS No. 86, Accounting for the Costs of Computer Software to be Sold, Leased, or Otherwise Marketed. SFAS No. 86 specifies that costs incurred internally in researching and developing a computer software product should be charged to expense until technological feasibility has been established for the product. Once technological feasibility is established, all software costs should be capitalized until the product is available for general release to customers. Judgment is required in determining when technological feasibility of a product is established. We have determined that technological feasibility for our software products is reached shortly before the products are released to manufacturing. Costs incurred after technological feasibility is established are not material, and accordingly, we expense all research and development costs when incurred."¹⁹

The R&D tax treatment of software is not substantially different from GAAP in this instance. Microsoft likely expenses this entire R&D under Section 174 for tax purposes as well. Section 174(a)(1) allows a taxpayer to treat R&D expenditures as expenses for income tax reporting purposes. Taxpayers also may choose to amortize certain R&D expenditures under Section 174(b)(1) ratably over a period of not less than 60 months. Many taxpayers are not likely to amortize software under Section 174(b)(1) because it reduces software expense and accelerates income tax payments they otherwise could defer, which destroys company value. Further, Section 174(e) states that only reasonable research expenditures are eligible under the circumstances.

The issue is that a substantial portion of software development for commercial products does not entail R&D. There are likely many R&D components in commercial software, but software programmers do not develop mature software projects of any meaningful size and scope purely using R&D methods—such products would never last on the market because of excessive quality control problems. In fact, a noted software development and process expert makes the following observation regarding research-oriented development activity:

"If your project strains the limits of computer science by requiring the creation of new algorithms or new computing practices, you're not doing software development; you're doing software research. Software-development schedules are reasonably predictable;

The issue is that a substantial portion of software development for commercial products does not entail R&D

¹⁹ *Id.*, page 37.

software research schedules are not even theoretically predictable."²⁰

Software development is like any other product development paradigm, in that there is a formal development process. R&D constitutes a small portion of this overall development paradigm, where proof-of-concepts are developed and tested. Once proof-of-concepts are complete, developers integrate lessons learned into production code. This is how professional software development occurs in practice regardless of whether developers use waterfall, spiral, object-oriented, extreme, rapid, or other software development methodologies.

Microsoft, for example, mentions localization in its annual report as an R&D activity. "Localization" means modifying software programs to use language and user-interface features native to a given country or locale. Thus, when Microsoft localizes Microsoft Windows XP for Spain, it modifies all menus, error messages, and other user-interface and program features to use Spanish text instead of English text. This hardly qualifies as technologically risky and is routine activity in the software development industry. Yet, Microsoft classifies this and likely thousands of other routine tasks in developing its products as R&D and expenses them. Further, by inference to the assertion made by the software development expert quoted above, if Microsoft were performing only R&D as he describes, Microsoft would never get a product out the door. Other examples of routine programming tasks within the Windows XP product that hundreds of software companies routinely employ include automatic updating, media players, web browsers, communication stacks, and device drivers. All are well-known and defined concepts in the industry. Yet Microsoft and other taxpayers categorize all costs associated with such development as R&D instead of capitalizing these costs. The wholesale expensing of all of these costs under Section 174(a)(1) should fail because of Section 174(e), which tests such expensing for reasonableness.

One needs only to look to R&D tax credit cases under Section 41 to see the clarity of what truly constitutes software R&D for tax purposes. For example, in *Eustace*, 90 AFTR 2d 2002-7661, 312 F3d 905 (CA-7, 2002), the court found that "normal software development" did not qualify for a tax credit under Section 41, and cited several cases to make its point:

"Taxpayers who have sought a credit under §41 for commercial software development have been uniformly unsuccessful. See, e.g., Tax & Accounting Software Corp. v. United States, 90 AFTR 2d 2002-6107, 301 F3d 1254 (10th Cir. 2002); Wicor, Inc. v. United States, 88 AFTR 2d 2001-5474, 263 F3d 659 (7th Cir. 2001); United Stationers, Inc. v. United States, 82 AFTR 2d 98-7488, 163 F3d 440 (7th Cir. 1998); Norwest Corp. v. CIR, 110 TC 454 (1998). The only exception—on which the taxpayers in this case principally rely—is the district court's decision in Tax & Accounting, 86 AFTR 2d 2000-5752, 111 F Supp 2d 1153 (N.D.

The wholesale expensing of all of these costs under Section 174(a)(1) should fail because of Section 174(e)

²⁰ See McConnell, *Rapid Development* (Microsoft Press, 1996), page 47.

Okla. 2000), which the tenth circuit reversed after they filed their brief. Applied Systems' claim is no better than the others.

"The evidence at trial in the Tax Court shows that Applied Systems engaged in normal software development.... None of it was pioneering; all of it entailed variations on themes long used by other developers. The Commissioner conceded that this work met many of the requirements in §41 but contested several others, of which only two require attention: that the research be 'undertaken for the purpose of discovering information which is technological in nature' (§41(d)(1)(B)(i)), and that the activities 'constitute elements of a process of experimentation' (§41(d)(1)(C)). The Tax Court concluded that Applied Systems flunked both tests—the former because it did not produce an 'innovation in underlying principle' and the latter because the research in question was not designed to dispel uncertainty about the technological possibility of developing software of this kind. T.C. Memo 2001-66 (2001)."

In the context of Eustace, much of the software Microsoft generates is likely routine and not R&D. As such, Microsoft should be capitalizing what may amount to a significant portion of its current R&D costs. Though Microsoft may follow the letter of the law for book and tax reporting purposes, current guidelines do not coincide with case law. Further, current guidelines violate the matching principle and distort Microsoft's true financial picture for book and tax purposes with overstated expenses and understated earnings.

If Microsoft does not expense all software under Section 174, then what provides guidance for it to treat development costs for tax reporting purposes? Section 263A appears to contain a reasonable set of guidelines. Under that section, if the taxpayer incurs most of its expenses before the software ever reaches the market, then it capitalizes all of the costs and deducts them ratably over some discrete useful life. In the context of Windows XP, assume Microsoft had \$100 million in true R&D for that product. Microsoft would capitalize its \$900 million in development costs and then expense them ratably over a useful life of something like four to six years. Assuming a five-year deduction period, Microsoft would incur R&D expenses in year 1 of \$280 million²¹ instead of \$1 billion, a difference of \$720 million. Microsoft would amortize the remaining \$720 million over years 2 through 5 at \$180 million per year. In 2001, when Microsoft launched Windows XP, Microsoft had a reported net income of \$7.346 billion.²² Had Microsoft capitalized and amortized the Windows XP development expense, its net income would have been about \$8.066 billion, or about 10% higher. That is just one product. Imagine similar proportional treatment (28% of the claimed amount as demonstrated

In the context of Eustace, much of the software Microsoft generates is likely routine and not R&D

²¹ \$100 million in pure R&D and \$180 in ratable deductions, or about 28% of the claimed amount.

²² See Microsoft Corporation 2001 Annual Report, page 38, available at www.microsoft.com/msft/ar.mspx.

above) across Microsoft's entire product line for the \$4.379 billion it spent on R&D. Microsoft's net income would increase some \$3.152 billion, or about 43%.

Section 263A sounds like an ideal place to address the issue of capitalizing and deducting software. Nevertheless, this section makes no explicit provision for software or for other intangibles in general (e.g., patents). Section 263A(b) does explicitly include certain intangible types such as "film, sound recording, video tape, book, or similar property." Construed narrowly, software does not apply in this context and Congress would have to make any change to the Code. Construed broadly, software would fall under "or similar property" and would be subject to capitalization under Section 263A. Notable is that software has a history of comparisons to the film and music industries in case law, so such broad interpretations are possible.²³

The software industry no doubt will object to any such changes regardless of how they occur because it will increase taxes due in earlier years of the product's life cycle. Substance should prevail over form, however, for internally developed software development projects, as it also should for purchased software projects. Both the FASB and tax jurisdictions should revisit this issue for internally developed software.

Like internally developed software, the tax treatment of embedded software is unduly complicated and depends on the tax type

EMBEDDED SOFTWARE—CURRENT TAX VIEW

Like internally developed software, the tax treatment of embedded software is unduly complicated and depends on the tax type (personal property, sales and use, or income) and jurisdiction (federal, state, or local). Each tax jurisdiction may apply different standards and tests to determine the taxability of software. Most are inconsistent and do not reflect the reality of the situation—they fail substance over form reasoning. This issue potentially has an even greater impact for taxpayers with the publication of final Regulations under Section 199. The probability of disputes between the IRS and taxpayers in this area is high because of the Service's inconsistent treatment of software to date.

Income Taxes

At the federal level, guidance for the tax treatment of software is found in Sections 167 and 197. Section 167(a) provides, as allowable, that taxpayers are entitled to a depreciation deduction of a reasonable amount for exhaustion, wear and tear, and obsolescence. Under Section 167(f)(1)(A), any such depreciation is to be computed using the straight-line method over a 36-month useful life. Section 167 specifically excludes software amortized in accordance with Section 197.

Taxpayers generally exclude embedded software from treatment under Section 197 because the products where vendors embed such software are available to the public, contain a non-exclusive license, and the vendor does not modify such

²³ See *Commerce Union Bank v. Tidwell*, 538 SW2d 405 (Tenn., 1976); *Universal Computer Associates*, *supra* note 5; *State of Alabama v. Central Computer Services, Inc.*, 349 So 2d 1160 (Ala., 1977).

software for each buyer. For example, when a vendor sells an ultrasound scanner, it sells each buyer the same scanner, running the same standard base software set. Further, purchase agreements generally forbid the buyer from reselling or reverse-engineering the ultrasound software. Tax treatment for such software therefore falls under Section 167.

Section 197, which addresses the amortization of goodwill and certain other intangibles, allows the taxpayer to amortize the software ratably over 15 years. Exceptions to the definition of intangibles, in Section 197(e)(3), exclude computer software available for purchase by the general public, which is subject to a non-exclusive license, and has not been modified substantially. In addition, this provision also excludes computer software that "is not acquired in a transaction (or series of related transactions) involving the acquisition of assets constituting a trade or business or substantial portion thereof."

Taxpayers generally exclude embedded software from treatment under Section 197 because the products where vendors embed such software are available to the public, contain a non-exclusive license, and the vendor does not modify such software for each buyer. For example, when a vendor sells an ultrasound scanner, it sells each buyer the same scanner, running the same standard base software set. Further, purchase agreements generally forbid the buyer from reselling or reverse-engineering the ultrasound software. Tax treatment for such software therefore falls under Section 167.

The IRS set forth guidance for treating software under Sections 167 and 197 in Rev. Proc. 2000-50, 2000-1 CB 601. Section 5 of that Procedure addresses the issue of at-risk developed software, which a taxpayer may expense under Section 174. Section 6 limits the Procedure's scope to acquired computer software subject to Section 167; it does not apply to software amortized under Section 197. The method the Service prefers for establishing embedded software value for income tax reporting purposes is the vendor invoice. The following language is found in section 6:

"With respect to costs of acquired computer software, the Service will not disturb the taxpayer's treatment of:

"(1) Costs that are included, without being separately stated, in the cost of the hardware (computer) if the costs are consistently treated as a part of the cost of the hardware that is capitalized and depreciated; or

"(2) Costs that are separately stated if the costs are consistently treated as capital expenditures for an intangible asset the cost of which is to be recovered by amortization deductions ratably over a period of 36 months beginning with the month the software is placed in service, in accordance with the rules under section 167(f)(1)."

The method the Service prefers for establishing embedded software value for income tax reporting purposes is the vendor invoice

State regulations generally mirror the federal rules for income tax reporting purposes. Thus, taxpayers generally can claim state income tax benefits for software only if the vendor states the software's value on its invoice. If the vendor does not state the software costs, the taxpayer loses the tax benefits. States generally are more flexible in their tax treatment of software with respect to tangible personal property taxes.

There is a side effect to the income tax treatment of software as described above. In many instances, the software has a much shorter useful life than what a tax jurisdiction mandates. For example, consider the case of a modern power-generation turbine system, which has a sophisticated embedded software-driven control system. The MACRS depreciation rate for this property is 20 years.²⁴ The problem is that vendors may not stock replacement part inventories for the control systems for 20 years; technology changes too quickly. Thus, power plants must carry an inventory of spare parts or they must replace the control systems long before the mandated useful life, only reinforcing that the useful life for the software and control system is overstated. This understates the software-related depreciation expense for a given period and overstates income, especially for equipment depreciated over a longer time with significant software content. Functional obsolescence—where a taxpayer incurs excess operating costs to keep the legacy software systems operational through additional staffing requirements, excessive salary for specialized staff skill sets, or lost efficiency—puts downward pressure on an asset's useful life as well.

Some tax jurisdictions point to instances where software is in use longer than the three-year life mandated by Section 167(f)(1)(A) in order to make a case for not accelerating the software expense for embedded software. This does not change the fundamental fact that software's appropriate life may be 36 months on average. And consider that many companies continue to own their furniture for many years after its seven-year class life: Tax jurisdictions have not sought to increase the class life of chairs because a taxpayer may own a given chair for 20 years.

Personal Property Taxes

The embedded software valuation issue is much more complex for personal property tax purposes than income tax purposes, for many reasons. First, not all states levy a personal property tax. Second, of the states that do levy a personal property tax, not all offer an exemption for software.²⁵ Third, of the states that do offer a personal property tax exemption for software, each has its own view of how to establish software value, and what software qualifies. Fourth, the status of these laws changes on what seems to be a daily basis as new case law develops, and

The embedded software valuation issue is much more complex for personal property tax purposes than income tax purposes

²⁴ See IRS Publication 946, How To Depreciate Property, Asset Class 49.13, page 106: "Includes assets used in the steam power production of electricity for sale, combustion [sic] turbines operated in a combined cycle with a conventional steam unit and related land improvements. Also includes package boilers, electric generators and related assets such as electricity and steam distribution systems as used by a waste reduction and resource recovery plant if the steam or electricity is normally for sale to others."

²⁵ See Banham, *supra* note 2.

legislatures maneuver to make their states the most attractive for high-tech investment.

The first issue to address at the state level is whether the software is exempt. This will vary by state. The tangible vs. intangible nature of the embedded software is paramount because property taxes generally apply only to tangible personal property. If a tax jurisdiction does not consider software intangible and exempt from personal property taxes, then the valuation issue for personal property tax purposes is moot. Some states have issued personal property tax regulations that mirror what is common in sales and use tax regulations. These jurisdictions (Oklahoma, for example) consider prewritten or "canned" software to be tangible and taxable, while software that is custom developed for taxpayers is considered intangible and exempt from personal property taxes. Custom software in this situation receives the more favorable tax treatment.²⁶

Unfortunately, Oklahoma is wrong because this treatment ignores the fact that prewritten software is still intangible. How many copies a vendor sells is incidental to the substance of the transaction. Consider this example: A retail supermarket chain contracts with a vendor to produce a custom communications device for it to display customer promotions in checkout lanes. The vendor creates a custom hardware and software package and delivers 500 copies of it to the supermarket's stores. In this case, the substance of the software is that it is definitely custom-developed for the supermarket—the vendor developed the project for the exclusive use of the supermarket. In form, however, it looks like a canned purchase because the vendor sells multiple copies of the software to the same buyer.

Also notable about Oklahoma's treatment is that it exactly the opposite from the Service's position, which provides more favorable tax treatment (a three-year amortization period) to prewritten software under Section 167 than to custom software (a 15-year amortization period) under Section 197.

The next issue in the personal property taxation of software is what is exempt and what is not. Many states tend to differentiate between operating system software and application software, the latter being anything that runs on top of an operating system.

Indiana, for example, states that the "operational program controls the hardware and actually makes the machine operational. It is fundamental and necessary to the functioning of the computer hardware itself and performs such functions as loading, scheduling, supervision, and data management."²⁷ Indiana defines application software as "a written sequence of instructions which details the operations the equipment is to perform in order to achieve a specific objective of the user."²⁸ Indiana would thus treat something like Microsoft Windows XP as an operating system and tax it as tangible personal property. Any software that runs

Many states tend to differentiate between operating system software and application software, the latter being anything that runs on top of an operating system

²⁶ See Kutten, *Personal Property Taxation of Computer Software: A State-by-State Guide* (4th ed., Software Taxation Institute, 1999-2000), page 194.

²⁷ 50 Ind. Admin. Code 4.2-4-3 (g) (2).

²⁸ 50 Ind. Admin. Code 4.2-4-3 (g) (3).

on top of Microsoft Windows XP, which is a common embedded operating system, is application software.

California has a narrower and perhaps more technically correct definition of operating system software. California would treat the Basic Input Output System (BIOS) of a modern desktop computer as the operating system. California considers everything else (including something like Microsoft Windows XP) as application software.²⁹ California is technically correct because a user can boot a computer without something like Microsoft Windows or Linux installed and the computer is still functional and usable. The same computer is useless without a BIOS.

Each state, however, ignores the fact that software is intangible regardless of function. To make software tangible in certain cases is arbitrary and ignores the fact that software developers create each software program, whether it is an operating system or Microsoft Word, in substantially the same manner.

The IRS has the most pragmatic view of software. Regardless of function or type, all software—from operating systems and executive code to application software and the tools used to build software (e.g., compilers & assemblers)—is treated the same.³⁰

Once the taxpayer identifies the exempt status of the subject software, the taxpayer then must determine the software's value. This is yet another source of confusion because the evidence that a taxpayer can use to establish the value of embedded application software varies by jurisdiction. States use two primary methods to establish software's value:

(1) The same valuation methodology used by the IRS—the vendor invoice. Software is valued at the cost stated on an invoice. If the cost is not stated, then the taxpayer loses the software exemption. Michigan, for example, uses this policy.

(2) An appraisal by an independent software valuation professional. For many states, this is still uncharted territory because the practice of segregating software for tax reporting purposes using an independent appraisal is a recent concept. Taxpayers further complicate this issue due to their abuses in various tax jurisdictions that do offer an exemption for embedded software. In two cases, the taxpayers attempted to exempt software without the use of an appraisal or independent valuation.³¹ Instead, they made random guesses to the software's value and did not follow statutory procedures for properly recording the software value. They lost. Worse, these taxpayers made it difficult for honest taxpayers to use professional,

The IRS has the most pragmatic view of software...regardless of function or type, all software...is treated the same

²⁹ See Kutten, *supra* note 26, at pages 60-61.

³⁰ Rev. Proc. 2000-50, 2000-1 CB 601, section 2.

³¹ See two Indiana tax cases: *Standard Plastic Corp. v. Department of Local Government Finance*, 773 NE2d 379 (Ind. Tax Ct., 2002), and *Autoliv North America v. Department of Local Government Finance*, 784 NE2d 593 (Ind. Tax Ct., 2003).

independent appraisals to establish the value of embedded software because of bad legal precedents.

FALLACIES OF PURE INVOICE REASONING

Tax jurisdictions commit administrative error when they rely solely on vendor invoicing to establish value for purchased software. First, there are established ways to determine value when the invoice does not state it. Second, the value of software has little to do with the incidental recording of the transaction on an invoice, which a vendor may understate deliberately.³² Substance over form should be the guiding principle, but tax jurisdictions ignore the argument. Third, most statutes never explicitly state the mechanism to establish software value, so it is definitely an administrative error. For example, nowhere in Section 167 does it state that the taxpayer must rely solely on a vendor invoice to establish value for software.

Return to the simple case of a taxpayer buying a new Dell computer loaded with Microsoft Office for \$1,000. Because Dell does not separately state the cost of Microsoft Office on its invoice, under Rev. Proc. 2000-50 the taxpayer loses the opportunity to expense the \$200 software over three years instead of five. The book cost to amortize over 36 months is zero. The book cost to depreciate over five years is \$1,000. In many states, the taxpayer also loses the ability to exclude the asset from personal property tax returns.

To see the fallacy in the invoice-only valuation argument, consider the same example modified slightly. Dell sells a taxpayer Microsoft Office for \$1,000 and includes a computer in the transaction. Dell never itemizes the hardware cost on its invoice. In accordance with Rev. Proc. 2000-50, the taxpayer amortizes the \$1,000 cost of the software over 36 months. The book cost of the hardware to depreciate over five years is zero. The taxpayer also has a zero basis for personal property tax purposes. By the letter of the law, the taxpayer has done everything correctly. The problem is that the IRS and other tax jurisdictions would seriously object if they saw this treatment, especially for larger dollar value items. Each would argue the substance of the transaction, which includes both hardware and software components—each with ascertainable value. The tax jurisdiction would likely assign some value to the "free" hardware using standard valuation methodologies (e.g., market comparables, scrap value). The hardware in this situation does have value regardless of what the invoice states.

Nevertheless, tax jurisdictions ignore the substance of the transaction for segregating software value from the hardware value when they require taxpayers to rely solely on a vendor invoice. Why? The software has ascertainable value even if the vendor does not state it. Moreover, the business purpose for software does not change based on incidental reporting of the vendor invoice. There is yet another wrinkle. If the value of the transaction is 51% or more related to software, and the

***Tax jurisdictions
commit administrative
error when they rely
solely on vendor
invoicing to establish
value for purchased
software***

³² See Armstrong, "Leverage Tax Exemptions for Embedded Software," McLean Report—Research Note (Info Tech Research Group, 2005).

taxpayer claims the entire balance as software, would the tax jurisdictions argue that the substance of the transaction is hardware related? They certainly argue the other side of this, where the substance of a transaction that is 51% or more related to hardware is entirely a hardware purchase.

It is clear that this is an area that needs review in certain tax jurisdictions. The good news is that the IRS and others have good guidance from other areas of the tax system, such as in real estate cost segregation, and from some states that have already updated their regulations. As shown in case law, tax jurisdictions must accept alternative valuation methods to establish value when segregating an asset into different value components for income tax reporting purposes.³³ The following comes from the Service's Cost Segregation Audit Techniques Guide (ATG) on the accepted practice of segregating lump-sum assets:

"In order to calculate depreciation for Federal income tax purposes, taxpayers must use the correct method and proper recovery period for each asset or property owned. Property, whether acquired or constructed, often consists of numerous asset types with different recovery periods. Thus, property must be separated into individual components or asset groups having the same recovery periods and placed-in-service dates in order to properly compute depreciation.

*"When the actual cost of each individual component is available, this is a rather simple procedure. However, when only lump-sum costs are available, cost estimating techniques may be required to 'segregate' or 'allocate' costs to individual components of property (e.g., land, land improvements, buildings, equipment, furniture and fixtures, etc.)."*³⁴

...standards and valuation methods developed in the Service's cost segregation audit manual coincide neatly with established appraisal valuation methods published for embedded software

In fact, standards and valuation methods developed in the Service's cost segregation audit manual coincide neatly with established appraisal valuation methods published for embedded software. Embedded software valuations developed and reported in accordance with the Uniform Standards of Professional Appraisal Practice (USPAP; more on this below) meet or exceed what constitutes a quality work product as discussed in chapter 4 of the ATG.

The function test. The next administrative error that tax jurisdictions commit regarding the tax treatment of software relates to the software's utility, with a "function test." Such discussions usually fall along the lines of "well, the machine would not work without the software." For example, a tax jurisdiction may argue that the software embedded in a retail point-of-sale system is crucial to the point-of-sale system's being operational, so the hardware and software are inseparable. This is not technically correct. The hardware works just fine. It operates and does everything its designers meant for it to do. It may not work as efficiently as

³³ See *Hospital Corp. of America*, 109 TC 21 (1997).

³⁴ See "Cost Segregation Audit Techniques Guide, Chapter 1—Introduction," available at www.irs.gov/businesses/article/0,,id=134122,00.html.

intended because the software optimizes the performance and functionality of the equipment, but the hardware still works. If the hardware did not work, the taxpayer would have to replace the hardware (e.g., an electrical storm fried the circuits).

In addition, if one looks to the other side of the argument, the fallacy of this view is again obvious. First, software sold separately does not "work" without hardware; however, tax jurisdictions do not increase the value of software to account for hardware necessary to run the software where the taxpayer purchases the software separately from hardware. In addition, tax jurisdictions like the IRS do not distinguish between software types in their regulations—software function is immaterial for their purposes.

Moreover, tax jurisdictions contradict their own vendor-invoice valuation method with a function test because regulations that rely on vendor invoices to establish value do not make any distinction as to software function. Consider again the retail point-of-sale system. The vendor itemizes the software cost on an invoice. Even though the software may not "work" in the eyes of a tax jurisdiction without the hardware (i.e., it fails the tax jurisdiction's presumed "function test"), the taxpayer is in full compliance if the taxpayer records the value of the software in books and records as stated on the vendor invoice. In fact, in full compliance with published regulations, the taxpayer rightly never considers software function in this process.

The function test becomes even more complicated than this. For example, software has value in printed form, never installed in a computer (i.e., a book of published software algorithms or design patterns). Publishers make millions selling books with these algorithms. This reinforces the argument that the medium and function are irrelevant.

The software function argument that tax jurisdictions may offer thus conflicts with their own regulations and it is nonsensical. A function test also conflicts with the intangible nature of software, where all software is intangible regardless of function, as well as the knowledge test as applied in prior case law.

THE VALUATION ISSUE

As stated previously, tax jurisdictions generally have two ways to establish the value of embedded software.³⁵ The vendor invoice is the first valuation approach and already has been discussed. The second, emerging way that tax jurisdictions establish value for embedded software is through an independent, third-party appraisal. Appraising software for tax purposes in a general sense is not necessarily something new. Companies evaluate internally developed and capitalized software costs routinely for impairment analysis, transfer pricing, and other tax matters.³⁶

Appraising software for tax purposes in a general sense is not necessarily something new

³⁵ See Pellegrino, "Cutting Taxes for Electrical Manufacturing and Coil Winding Equipment Companies," Proceedings of the 2005 EMCW Conference (EMCW, 2005), page 264.

³⁶ See Pellegrino, *supra* note 8.

Numerous references have documented the general approach for valuing software, both for embedded software and internally developed software.³⁷ Tax jurisdictions have used these standards for validating taxpayer claims for software.³⁸ This is good because industry acceptance of the valuation methods is crucial to valuation credibility. In fact, the IRS uses software valuation methods described in one reference book for its audit efforts to validate taxpayer write-off claims.³⁹ Valuations developed and reported in compliance with USPAP bring additional credibility.

As already mentioned, USPAP requirements for establishing value are perhaps even more stringent than what is currently required by the IRS in other tax matters, such as for real-estate cost segregation studies. In fact, the ATG readily admits to few standards.⁴⁰ Both the ATG and USPAP standards require demonstrated appraiser competence. USPAP compliance does not allow for contingent-fee arrangements, however, which are common in the cost segregation industry. USPAP compliance thus keeps aggressive taxpayers and consultants at bay. Also, USPAP compliance requires substantial appraisal development and reporting. The ATG mentions "rules of thumb" as a possible valuation method, which do not pass muster under USPAP appraisal development standards. Further, courts recognize USPAP usage; USPAP has legal precedent; USPAP has tests appraisers must pass on conduct, ethics, appraisal development, and appraisal reporting; and the Financial Institutions Reform, Recovery, and Enforcement Act of 1989 (FIRREA) recognizes USPAP as the generally accepted appraisal standard and requires USPAP compliance for appraisers in federally related transactions.

Of course, when the software tax issue first emerged, USPAP and software valuation methodologies were not yet around. Much has happened in this arena since, however, and it is time to update tax regulations to match. Some states are already doing this.

PROGRESSIVE THINKERS

Certain states have recognized the fallacy of vendor invoice reasoning and have changed their laws accordingly. Indiana is the latest example of a jurisdiction that recognizes acceptable alternative valuation methods to a vendor invoice for personal property tax reporting purposes. In fact, Indiana enacted legislation in 2005 to allow for the segregation of software from hardware costs, particularly when the vendor does not state the value of software on an invoice. The state's focus is on economic development. Indiana's legislature and governor want to attract high-tech investment and jobs. They created an economic climate to help stimulate that.

USPAP requirements for establishing value are perhaps even more stringent than what is currently required by the IRS in other tax matters

³⁷ See Reilly and Schweihs (eds.), *The Handbook of Business Valuation and Intellectual Property Analysis* (McGraw-Hill, 2004); Reilly and Schweihs, *Valuing Intangible Assets* (McGraw-Hill, 1999); Kuttan, *Personal Property Taxation of Computer Software: A State-by-State Guide*, 4th ed. (Software Taxation Institute, 1999-2000); Pellegrino, *supra* note 8.

³⁸ See Pellegrino, *supra* note 8.

³⁹ See Boehm et al., *Software Cost Estimation with COCOMO II* (Prentice-Hall, 2000), page 1.

⁴⁰ See note 34, *supra*.

Indiana was careful about how it designed the rules; the state addressed two key issues—generating valuation results that are both equitable and credible. Indiana regulations require taxpayers to use published software valuation methodologies and an independent appraisal to assign software value. Taxpayers do not make their own value determinations—they generally are not competent to do so under appraisal development standards. Even if they were competent, taxpayers have a conflict of interest in rendering their own value opinions—they have a vested interest in the outcome, which is in effect a contingent fee arrangement.

Indiana's pending regulations on the appraisal of software are as follows:

(i) The true tax value at the time of acquisition of computer application software may be determined using the following:

(1) An independent, professional appraisal:

(A) must be made in conformance with generally accepted standards for appraisal practice;

(B) shall not be based on a contingent fee arrangement;

(C) shall include consideration of the cost, market, and income approaches; and

(D) shall distinguish the boundary in the equipment between exempt intangible application software and nonexempt tangible operational software.

The appraiser must have demonstrated competence in the valuation of software.⁴¹

Indiana was careful about how it designed the rules...generating valuation results that are both equitable and credible

By design, Indiana's new legislation is especially meaningful for companies considering locating high-tech equipment in the Midwest. For example, if a manufacturing company is considering locating equipment with much embedded software in Michigan or Indiana, it may be better off in Indiana since Indiana has more favorable treatment for software for personal property tax purposes. Michigan follows the invoice-only method of establishing value. A \$10 million investment in seven-year equipment with a 30% software content and a 3% tax rate, all else the same, will cost the company over \$500,000 more if it chooses Michigan over Indiana.⁴²

THE SAVINGS EXAMPLE

In light of the above discussion, we will illustrate how these savings may accumulate for a taxpayer. For the purposes of this example, assume the following conditions:

⁴¹ See www.in.gov/legislative/register/xml/old-ir/Vol29/08May/08P050050252.PDF.

⁴² Savings calculated in accordance with true-tax valuation procedures as outlined in 50 Ind. Admin. Code 4.2-4-7. Annual property tax reductions are \$84,000, \$117,600, \$88,200, \$67,200, \$50,400, \$37,800, \$31,500, and \$31,500, respectively.

- (1) The equipment is located in Indiana, which exempts embedded application software from business personal property taxes and allows for a professional appraisal to determine the software's value.
- (2) The equipment costs \$1 million.
- (3) The software content valuation is 50% of the equipment's cost.
- (4) The taxpayer depreciates the equipment over a seven-year useful life.
- (5) The state personal property tax rate is 3%.
- (6) Taxpayer has an effective income tax rate of 40%.
- (7) A discount rate of 15% is used to account for the time value of money.

... the present value of the combined state and federal tax benefits accumulate to \$48,198, or almost 5% of the equipment's purchase price

Exhibit 1 demonstrates how the present value of the combined state and federal tax benefits accumulate to \$48,198, or almost 5% of the equipment's purchase price. About half of that benefit comes from increased cash flow from faster expensing of the capitalized software. Before segregating the software cost, the present value of the depreciation tax benefit was \$250,979. After restructuring the transaction, the present value of the depreciation and amortization tax benefit increased to \$276,182. In total, the taxpayer receives a \$25,204 benefit by deferring income taxes.

The balance of the savings comes from outright reductions in property taxes. In Indiana, embedded application software is exempt from personal property taxes, so the taxpayer reduces its property tax basis by \$500,000 per year. That additional reduction has a present value tax savings of \$22,944.

As stated earlier, an interesting side effect of not segregating the software properly distorts the company's earnings due to an underreported software expense, and can widen the gap between book and tax value. This is clear in the example above. Before software segregation, the taxpayer incurs a depreciation expense of \$140,000 for the year. After segregation of the assets, the taxpayer incurs a combined depreciation and amortization expense of \$235,000. The accelerated expense better reflects the reality of the situation: the software will not likely operate in the same form over the life of the equipment. The taxpayer will incur additional cost for software maintenance and upgrades.

CONCLUSION

Vendors embed software in a substantial amount of equipment that they sell. This embedded software enables many product features that vendors historically implemented with electrical or mechanical systems. This embedded software is entitled to favorable tax treatment to the extent that the taxpayer can establish the software's value. This includes faster expensing of the software for income tax purposes and a reduction in personal property tax in states that allow for a straight

Tax Analysis, Before Software Segregation

	<u>Year 1</u>	<u>Year 2</u>	<u>Year 3</u>	<u>Year 4</u>	<u>Year 5</u>	<u>Year 6</u>	<u>Year 7</u>	<u>Year 8</u>
<u>Income Tax</u>								
MACRS 7 Year Depreciation Rate	14%	25%	17%	13%	9%	9%	9%	4%
Depreciation Expense	140,000	250,000	170,000	130,000	90,000	90,000	90,000	40,000
Income Tax Benefit	56,000	100,000	68,000	52,000	36,000	36,000	36,000	16,000
Income Tax Benefit Present Value	250,979							
<u>Property Tax</u>								
Indiana True Tax Value Percentage	40%	56%	42%	32%	24%	18%	15%	15%
Property Tax Payments	(12,000)	(16,800)	(12,600)	(9,600)	(7,200)	(5,400)	(4,500)	(4,500)
Present Value of Property Tax Payments	(45,989)							

Tax Analysis, After Software Segregation

	<u>Year 1</u>	<u>Year 2</u>	<u>Year 3</u>	<u>Year 4</u>	<u>Year 5</u>	<u>Year 6</u>	<u>Year 7</u>	<u>Year 8</u>
<u>Income Tax</u>								
MACRS 7 Year Depreciation Rate	14%	25%	17%	13%	9%	9%	9%	4%
Depreciation Expense	70,000	125,000	85,000	65,000	45,000	45,000	45,000	20,000
Income Tax Benefit	28,000	50,000	34,000	26,000	18,000	18,000	18,000	8,000
Straight-line 3 Year Amortization Rate	33%	33%	33%	0%	0%	0%	0%	0%
Amortization Expense	165,000	165,000	165,000	-	-	-	-	-
Income Tax Benefit	66,000	66,000	66,000	-	-	-	-	-
Total Income Tax Benefit	94,000	116,000	100,000	26,000	18,000	18,000	18,000	8,000
Income Tax Benefit Present Value	276,182							
<u>Property Tax</u>								
Indiana True Tax Value Percentage	40%	56%	42%	32%	24%	18%	15%	15%
Property Tax Payments	(6,000)	(8,400)	(6,300)	(4,800)	(3,600)	(2,700)	(2,250)	(2,250)
Present Value of Property Tax Payments	(22,994)							

Benefit of Software Segregation

Income Tax Benefit Increases	25,204
Property Tax Payment Reductions	22,994
Total Present Value of Benefits	48,198

Exhibit 1 Software Segregation Analysis

exemption. Historically, tax jurisdictions relied only on what a vendor stated on an invoice to establish embedded software value. If the vendor did not state the value, then the taxpayer lost the beneficial tax treatment for software.

Tax agencies and courts have conflicting laws, regulations, and precedents. A distinction between internally developed and purchased software exacerbates the issue. Taxpayers account for and exclude a disproportionate share of R&D activity related to routine production activities. Many tax jurisdictions misunderstand what software is and misapply various tests and analogies in determining how to treat software for tax purposes. In many instances, tax jurisdictions ignore substance over form rules and seek to maximize revenue. For income tax reporting purposes, tax jurisdictions continue to prefer using vendor invoices to establish value. For personal property tax reporting purposes, the valuation method varies from the invoice method to independent assessments by qualified appraisers. The software that is exempt at the state level varies by jurisdiction and each has its own definition of what qualifies. Tax jurisdictions that rely on vendor invoices

exclusively to establish software value commit administrative error. Tax jurisdictions use other valuation methods for lump-sum asset allocations in other tax areas such as in real-estate cost segregation studies routinely. Software valuation experts typically develop and report their work products in accordance with USPAP, which meets or exceeds valuation requirements for other tax matters. Certain states, such as Indiana, have updated their tax policies to allow for the segregation of software using an appraisal when a vendor does not state the software's value on its invoices. Taxpayers may stand to recoup 10% or more of the acquisition cost of their equipment through these cumulative tax benefits.

PRACTICE NOTES

Taxpayers must update their books and records to reflect the situations where they segregate software value from hardware value. Doing so provides continuity between a taxpayer's tax treatment and book treatment, and in some states (such as Indiana) there may also be a statutory requirement to do so. For example, consider a \$1,000 asset that has an embedded software value of \$400. Instead of booking the asset cost of \$1,000 attributable only to hardware, the taxpayer should book the total cost as two separate assets, one software asset for \$400 and one hardware asset for \$600.