

Open Source Software Licenses: Legal Implications and Practical Guidance

by Elsie DeBrie and David Goeschel

The authors would like to thank Matthew Gillespie, law clerk at Koley Jessen and third-year law student at Washington University School of Law in St. Louis, MO, for his assistance on this article. Matthew is finishing his third year of law school at the University of Nebraska College of Law.

The open source software model is a community approach to software development that implements non-traditional licensing arrangements to promote the free distribution of software and the human readable programming instructions (known as “source code”) underlying that software. In addition, users of open source software are permitted to create derivative works of that software for any purpose.¹

Open source software offers a number of attractive advantages to businesses. Because the source code for open source software is available to the public, many open source projects are developed, refined and improved upon by the developer community at large. A company that can make use of such software in its business may obtain the benefits of a thoroughly-tested program while saving the cost and time it takes to develop the software internally and avoiding substantial license fees associated with traditional software licenses. Further, the company can maintain and improve upon the software in-house, which it might be unable to do under a traditional software license agreement.²

While open source software provides many valuable benefits to businesses and the software development community, the licenses for open source software often contain stringent requirements. When companies use open source software in their products without understanding the terms of the applicable licenses, they may inadvertently incur risk related to the protection of their valuable intellectual property rights and, thereby, dilute the value of their software products and business.

The widespread use of open source software suggests that its practical implications are widely understood and appreciated. Unfortunately, the legal implications of using open source software are not yet widely known or understood. This article will: (1) briefly summarize the history of open source software; (2) provide an overview of open source software licenses and issues that arise with respect to compliance with their terms; (3) discuss the limited universe of available caselaw that provides guidance on the legal effect of open source software licenses; and (4) conclude by offering practical steps that companies can



Elsie DeBrie



Elsie DeBrie is an Associate at Koley Jessen P.C., L.L.O. She drafts and negotiates a variety of commercial and technology agreements and advises clients on the protection of their intellectual property rights. Elsie obtained her B.A. from Doane College (magna cum laude) and her J.D. from the University of Virginia College of Law.

David Goeschel



David Goeschel is a Shareholder at Koley Jessen P.C., L.L.O. He counsels clients on the protection and commercialization of intellectual property assets, with a focus on structuring and negotiating a wide range of commercial and technology agreements. David obtained a B.A. and B.S. from the University of Kansas (with honors) and his J.D. from Creighton University School of Law (cum laude).

OPEN SOURCE SOFTWARE LICENSES

take to manage their risk with respect to the use of open source software in their businesses.

Historical Context

The origin of open source software traces back to the Massachusetts Institute of Technology (MIT) in the 1970s, where Richard Stallman worked as a computer programmer and was part of a community of programmers in which open sharing of software was the norm. In contrast to his experience in the academic setting at MIT, the reality of proprietary software in the commercial sphere was repugnant to Stallman. At one point, Stallman sought to improve the performance of a printer at MIT that frequently jammed, but he was unable to do so because the manufacturer refused to give him the source code for the printer driver. A colleague from Carnegie Mellon who had access to the source code was prevented from sharing it with Stallman because of a non-disclosure agreement with the manufacturer. Recalling the experience later, Stallman said, "This was my first encounter with a non-disclosure agreement, and I was the victim . . . non-disclosure agreements have victims. They're not innocent. They're not harmless."³

Over time, Stallman's community of programmers at MIT dissolved, and the increasing prevalence of NDAs and commercial software licenses greatly restricted the open atmosphere that Stallman had previously known. This, ultimately,

led Stallman to found the Free Software Foundation ("FSF") in 1984, a non-profit organization whose goal was to create and foster an open software community. The FSF's first step toward that goal was the development of a non-proprietary operating system, called GNU.⁴ The FSF licensed GNU under the first open source software license, called the General Public License (GPL). Stallman developed the GPL to secure for software users what the FSF calls the four essential freedoms: (1) the freedom to run the program, for any purpose; (2) the freedom to study how the program works, and adapt it to your needs; (3) the freedom to redistribute copies so you can help your neighbor; and (4) the freedom to improve the program, and release your improvements to the public, so that the whole community benefits.⁵

In February 1998, members of the software community held a strategy session in Palo Alto, CA, to consider ways to engage and encourage programmers to participate in community development of software and to emphasize the practical, business reasons for an open development process. It was at this strategy session where the term "open source" was coined. The term "open source" was chosen to create some distance from the FSF's politically and philosophically-driven "free software" term.⁶ Later that same month, two attendees at that strategy session, Eric Raymond and Bruce Perens, founded the Open Source Initiative (OSI). The OSI's initial goal was to

Landex Research, Inc.
PROBATE RESEARCH



**Missing and Unknown Heirs
Located
with No Expense to the Estate**

Domestic and International Service for:
Courts
Lawyers
Trust Officers
Administrators/Executors

1345 Wiley Road, Suite 121, Schaumburg, Illinois 60173
Telephone: 847-519-3600 Fax: 800-946-6990
Toll-Free: 800-844-6778
www.landexresearch.com

OPEN SOURCE SOFTWARE LICENSES

advocate for open software development and to encourage use of the term “open source.” The OSI later developed the Open Source Definition, which sets forth ten criteria that a software license must meet in order to qualify as open source.⁷

Today, the GPL is one open source software license among hundreds, whose terms and effect vary widely.⁸ These licenses are typically drafted by organizations that sponsor open source projects, but any developer that writes a computer program may choose to open source his work and license it under any open source software license he pleases. Since the development of GNU, the use of open source software, by developers and the business community alike, has dramatically increased. Widely-used programs such as Mozilla Firefox, Wordpress and Linux are among the ranks of open source software and are used by millions of people around the world. A recent survey indicates that as many as 78% of companies incorporate open source software into their own software products.⁹

Overview of Open Source Software Licenses

Open source software licenses typically fall into one of two broad categories: permissive (or non-copyleft) licenses and restrictive (or copyleft) licenses.

Permissive Licenses

Permissive licenses typically permit free and unrestricted use of the licensed software, as long as the user complies with certain minimal requirements, which typically include providing attribution when the user makes use of the author’s code in its own software, by including a copyright notice within the source code of the work. Some commonly-used permissive licenses are the MIT License, Apache License 2.0, and the BSD License.¹⁰ For businesses interested in selling proprietary software, permissive licenses are generally considered low risk, high reward.

Restrictive Licenses

As the name suggests, restrictive (copyleft) licenses contain much more onerous requirements than permissive licenses and present a greater degree of risk to companies that want to incorporate copyleft-licensed code into their products. Some commonly-used restrictive licenses are the GPL (versions 2 and 3), the Affero GPL (AGPL), the Lesser GPL (LGPL) and the Mozilla Public License 2.0 (MPL).¹¹ Note, however, that the LGPL and the MPL are considered “weak” copyleft licenses because their requirements are less restrictive than those of the GPL and the AGPL. Most of the rest of this article will explore the requirements of copyleft licenses and their legal implications, with particular focus on the GPL.

Copyleft licenses seek to realize some version of the FSF’s radical view of software freedom by invoking copyright law but

subverting its traditional goals. While copyright is traditionally concerned with protecting an *author’s* right to exclude others from copying, distributing, or modifying a work, copyleft seeks to strengthen the rights of *users* by prohibiting the imposition of limits on the right to copy, distribute, or modify a work.

In practice, copyleft licenses require the licensed work, as well as any work based on the licensed work (discussed below), to continue to be licensed under the same copyleft license. For example, all works that incorporate or are otherwise based on software licensed under the GPL must also be licensed under the GPL. This has been described as the “viral” nature of copyleft because copyleft-licensed code “infects” any code that it touches. In addition, a company that incorporates copyleft code into its product must provide the complete source code for its software product to all downstream recipients (i.e., to anyone to whom the company distributes an executable copy of its software product) and may not impose limitations on downstream recipients’ right to copy or modify the program.

The practical impact of a company’s use of copyleft software in its proprietary product, without proper consideration of the risks associated therewith, can be crippling. If copyleft obligations are triggered, and a company is required to release the source code underlying its proprietary software product, it may be difficult to commercialize its product and the value of its business may plummet. Importantly, the copyleft obligation to provide the source code for a program is only triggered once a derivative work of the copyleft work has been created, and the derivative work has been distributed. As such, the most important questions for a company wanting to avoid copyleft obligations are: (1) what constitutes a derivative work, and (2) what constitutes distribution?

What Constitutes a Derivative Work?

Under the United States Copyright Act, a “derivative work” is “a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, [etc.]”¹² The question of what this means in the context of software and copyleft licenses is open from a legal standpoint because courts have not directly addressed the viral provisions of the GPL or any other copyleft license. Moreover, the question is complicated from a technical standpoint because different parts of the software stack and different kinds of programs interoperate in different ways—plug-ins interact differently with a program than does a statically-linking library, which, in turn, interacts differently with a program than does a dynamically-linking library.¹³ This article will not discuss the technical aspects of the question, but it is worth noting that computer science experts disagree about the extent to which two programs need to interact and combine in order for a derivative work to be created and the viral provisions of the GPL to be triggered.¹⁴



OPEN SOURCE SOFTWARE LICENSES

The latest version of the GPL (GPLv3) recalls the language of the Copyright Act to define “covered work” as “either the unmodified Program or a work based on the Program.” Then, under the section of GPLv3 governing the distribution of derivative works, GPLv3 states that the license applies “to the whole of the work, and all its parts, regardless of how they are packaged.”¹⁵ The FSF and the Software Freedom Law Center (SFLC), which provides pro bono legal services to open source software developers, take an aggressive approach to the question of what constitutes a derivative work under the terms of the GPL.¹⁶ The key, from these organizations’ perspective, to distinguishing whether a work is a derivative work or a separate work appears to lie in determining whether the works communicate with each other when they run.¹⁷

To further illustrate the breadth of the FSF’s interpretation of derivative works, it is worth examining what it does *not* consider to be a derivative work. Section 5 of GPLv3 sets forth an “aggregate” as something that is not a derivative work and, thus, does not trigger the viral provisions of the GPL. According to the FSF website, “an ‘aggregate’ consists of a number of separate programs, distributed together on the same CD-ROM or other media. The GPL permits you to create and distribute an aggregate, even when the licenses of the other software are non-free or GPL-incompatible.”¹⁸

The consequences of the FSF’s radical approach is signifi-

cant: If a company’s proprietary software communicates with a tool that is licensed under the GPL when it runs, then it is a derivative work of that tool. Once a derivative work has been created, the entire work must be licensed under the same license as the original work. Then, once the derivative work is distributed, all users who receive binary or executable copies of the work are entitled to receive the corresponding source code for the work.¹⁹ As such, to comply with these licenses, all derivative works that are distributed become, essentially, nonproprietary.

What Constitutes Distribution?

Software-as-a-Service

The Software as a Service (SaaS) delivery model presents a method for companies to provide customers with their proprietary software without distributing it to them. Under a SaaS delivery model, customers access software hosted on a remote server via the Internet and never receive an executable copy of the software on their computers. As such, no distribution of the software occurs, and the copyleft requirement to provide the underlying source code for the software is not triggered. Many companies, including Google, have taken advantage of this “loophole” to great success by keeping GPL-licensed software in the cloud, thereby avoiding release of the source code for their products.

ALPS NEBRASKA

PROUD PARTNER OF THE NEBRASKA
LEGAL COMMUNITY



ALPS



- The nation’s largest direct writer of lawyers’ malpractice insurance
- Endorsed by more State Bars than any other carrier
- Most competitive pricing for the broadest coverage
- Financial Strength
- Founded by lawyers, for lawyers

(800) 367-2577 | www.alpsnet.com

OPEN SOURCE SOFTWARE LICENSES

In general, this loophole is well-established and widely exploited—indeed, GPLv3 explicitly blesses this loophole,²⁰ and the Affero GPL (AGPL) and other licenses were subsequently developed in order to close this loophole for those authors who want copyleft obligations to apply even in the SaaS context. Despite the general acceptance of the SaaS loophole, there are some contexts where a SaaS provider may unwittingly distribute its product and trigger the copyleft provisions attached to its code.

For instance, SaaS products may incorporate software tools that cause a portion of the software to execute on a user's browser. This execution of a copy may well constitute distribution. One law journal underlines the issue:

Courts have held that copying occurs when the information necessary to make a copy on a user's RAM is transferred, even if no actual material copy is transferred. Thus, the argument proposes that a copyrighted portion of software copied onto a user's RAM is sufficient to find distribution, even for [SaaS providers].²¹

This is illustrated through Mozilla's interpretation of the MPL. According to Mozilla, any "code which is sent to the client (e.g. HTML, CSS, JavaScript) does count as 'distributed.'"²² Thus, a SaaS product using JavaScript tools, for example, may well cause distribution of the product, thereby triggering the requirements of the applicable open source software license.

Mergers and Acquisitions

Another instance where distribution may occur is in the M&A context. When a SaaS company is sold, particularly in an asset sale, the transfer of the target company's assets to a buyer likely constitutes delivery or distribution of those assets. In such a case, distribution of the target company's copyleft software product to the buyer triggers copyleft obligations. While the courts have not ruled on this issue, some legal scholars have recognized this risk. One such scholar explains:

If a change of control is an assignment by operation of law, one might logically conclude that it also constitutes providing a copy to another entity, and thus a distribution triggering copyleft provisions. Keep in mind, also, that the effectuation of some forms of M&A transactions such as asset sales are clearly assignments, and also likely . . . constitute distribution[.]²³

Indeed, the FSF and SFLC take this position with respect to all acquisitions, stating, "in business acquisitions, whether by sale of assets or transfers of control, the acquiring party is downstream from the party acquired."²⁴ The obligation to provide the source code to an acquiring entity is not unduly burdensome in itself, but the fact that the entire software product is licensed under a copyleft license may have serious implications for the valuation of the target company.

Caselaw

There is little caselaw that speaks to the enforceability of copyleft licenses or fleshes out the requirements of their terms because most judicial actions that have been initiated in this context have been settled prior to trial. While the legal underpinnings of copyleft are not firm,²⁵ free software advocates have enjoyed enough success, through settlements, to suggest copyleft licenses have teeth.

One area of legal uncertainty lies in whether a violation of the terms of a copyleft license constitutes copyright infringement or breach of contract. The answer matters for purposes of determining available remedies and determining the enforceability of some of the terms of the licenses themselves.²⁶ If a license violation constitutes copyright infringement, then, provided the licensor has a federal copyright registration for its software, the licensor can sue in federal court, obtain an injunction from further infringement and seek statutory damages.²⁷ If, on the other hand, a license violation constitutes breach of contract, then the licensor's remedy will likely be limited to damages, though the licensor could also seek specific performance of the license. Specific performance, though seldom granted by courts in the U.S., could be a devastating remedy for a licensee, if the licensee were compelled to release the source code for its proprietary product.²⁸ Under the most prominent open source software case, *Jacobsen v. Katzer*, whether violation of an open source software license constitutes copyright infringement or breach of contract depends on the language of the license.²⁹

In *Katzer*, the major issue was whether the defendant's failure to comply with the Artistic License's attribution requirements constituted a breach of contract or copyright infringement. Ultimately, the Federal Circuit found the attribution requirement was a condition of the right to use the copyrighted work and not merely a contractual covenant.³⁰ Thus, defendant's conduct gave rise to a claim of copyright infringement. Beyond the bare holding, the court openly endorsed the enforceability of open source software licenses, stating:

Copyright licenses are designed to support the right to exclude; money damages alone do not support or enforce that right. The choice to exact consideration in the form of compliance with the open source requirements of disclosure and explanation of changes, rather than as a dollar-denominated fee, is entitled to no less legal recognition. Indeed, because a calculation of damages is inherently speculative, these types of license restrictions might well be rendered meaningless absent the ability to enforce through injunctive relief.³¹

This was a victory for copyleft licenses, but the later case of *MDY v. Blizzard* has the potential to sour that victory.³² *Blizzard* did not involve open source software licenses, but it touched on the issue of when breach of the terms of a license

OPEN SOURCE SOFTWARE LICENSES

gives rise to a copyright infringement action. In *Blizzard*, the Court held that, “for a licensee’s violation of a contract to constitute copyright infringement, there must be nexus between the condition and the licensor’s exclusive rights of copyright.”³³ One law journal article highlights the potential significance of this holding in the context of copyleft:

Since the Copyright Act grants only the rights to reproduce the work, prepare derivative works, distribute copies, perform the work publicly, and display the work publicly, it remains to be seen whether any of the common [copyleft] requirements of distribution of derivative works’ source code, application of an identical license to derivative works, notation of changes between the original and derivative works, and nondiscrimination would be held unenforceable.³⁴

More recently, in *Versata Software, Inc. v. Ameriprise Financial, Inc.*,³⁵ Versata (as licensor) sued Ameriprise for breach of their Master Software License Agreement because Ameriprise permitted third-party contractors to access and work on Versata’s proprietary software in violation of the agreement. When discovery revealed that Versata had incorporated GPL-licensed software owned by XimpleWare, Inc., into its software without complying with the GPL’s copyleft obligations, Ameriprise brought a counterclaim against Versata, arguing that Versata was required to provide the source code for its product to all users, including Ameriprise and its third-party contractors. Versata, then, argued that Ameriprise’s counterclaim was preempted by copyright law.

The court disagreed with Versata’s preemption argument, stating:

The “viral” component of the GPL is separate and distinct from any copyright obligation. Copyright law imposes no open source obligations, and Ameriprise has not sued Versata for infringing [a non-party’s] copyright by distributing [protected materials] without permission. Instead, Ameriprise has sued based on Versata’s breach of an additional obligation: an affirmative promise to make its derivative work open source because it incorporated an open source program into its software. Ameriprise’s claim therefore requires an “extra element” in addition to reproduction or distribution: a failure to disclose the source code of the derivative software. The presence of an additional contractual promise separate and distinct from any rights provided by copyright law means Ameriprise’s claim is not preempted.³⁶

The Court did not address *Katzer*’s covenant-versus-condition inquiry in the context of the GPL, but its reasoning is reminiscent of *Blizzard* and suggests that there is not a sufficient nexus between the exclusive rights of copyright, on the one hand, and the obligation under the GPL to make the source code of derivative works available to users, on the other hand, to yield a copyright infringement cause of action.

The Courts in *Katzer*, *Blizzard* and *Versata* have provided clues as to whether and how the terms of the GPL and similar copyleft licenses would be enforced, but we have yet to see a definitive answer in the law.

Practical Considerations

Attorneys should be aware of the issues involved with open source software and copyleft licenses, in particular, and stay apprised of new and ongoing litigation surrounding those issues, so that they can advise clients on the risks and uncertainties that underlie the incorporation of open source software into their proprietary products. Even given these risks and uncertainties, though, the concrete advantages that open source software offers to businesses should not be ignored.

For clients that would like the flexibility to incorporate open source software into their products, attorneys can encourage those clients to take few concrete steps in order to manage the risk they assume. For instance, companies can educate their personnel on the terms of the most-widely-used open source software licenses and how to comply with them. In addition, companies can develop open source policies that include a list of licenses that are acceptable to their business from a risk standpoint and a list of those that should be avoided. Such policies can also establish procedures that personnel are to follow when they encounter open source software they would like to implement within the company’s products, such as notification of appropriate management personnel and review of the terms of the applicable license. By educating personnel and implementing a simple open source policy, businesses may be able to “have their open source cake and eat it, too.” 

Endnotes

- 1 See generally Dennis M. Kennedy, “A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture,” at n.1, University of Miami, <http://www.cs.miami.edu/home/burt/learning/Csc322.052/docs/opensourcedmk.pdf> (accessed Feb. 1, 2016).
- 2 See Richard Raysman, “Open-Source Software: Use and Compliance,” Practical Law, <http://us.practicallaw.com/9-504-7111> (accessed Feb. 28, 2014).
- 3 David Bretthauer, “Open Source Software: A History,” at 4, UConn Libraries Published Works (Dec. 26, 2001), http://digitalcommons.uconn.edu/cgi/viewcontent.cgi?article=1009&context=libr_pubs.
- 4 *Id.*
- 5 “What is Free Software?,” Free Software Foundation, available at <http://www.gnu.org/philosophy/free-sw.en.html>.
- 6 “History of the OSI,” Open Source Initiative, available at <http://opensource.org/history>.
- 7 See “The Open Source Definition,” Open Source Initiative, available at <http://opensource.org/osd>.
- 8 Richard Kemp, “Open-Source Software,” at 7, Practical Law, <http://us.practicallaw.com/0-500-4366> (accessed Feb. 28, 2014).
- 9 North Bridge, Black Duck Software, “Seventy-Eight Percent of Companies Run on Open Source, Yet Many Lack Formal Policies to Manage Legal, Operational, and Security Risk” Black Duck Software (Apr. 16, 2015), <https://www.blackduckssoftware.com>.

OPEN SOURCE SOFTWARE LICENSES

com/news/releases/seventy-eight-percent-companies-run-open-source-yet-many-lack-formal-policies-manage.

¹⁰ These licenses are available on the OSI's website at <https://opensource.org/licenses/>.

¹¹ *Id.*

¹² 17 U.S.C. §101.

¹³ See Kemp at 15.

¹⁴ See *id.* at 19-20.

¹⁵ GPLv3 §5

¹⁶ See Eben Moglen and Mishi Choudhary, "Software Freedom Law Center Guide to GPL Compliance," 2d. ed., 6-8, (Oct. 31, 2014) available at https://www.softwarefreedom.org/resources/2014/SFLC-Guide_to_GPL_Compliance_2d_ed.html; "Frequently Asked Questions about the GNU Licenses," Free Software Foundation, available at <http://www.gnu.org/licenses/gpl-faq.en.html#MereAggregation>.

¹⁷ See *id.*

¹⁸ "Frequently Asked Questions about the GNU Licenses," Free Software Foundation, available at <http://www.gnu.org/licenses/gpl-faq.en.html#MereAggregation>.

¹⁹ See GPLv3 §6.

²⁰ GPLv3 §0 (in definition of "convey," the GPLv3 states: "mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.")

²¹ Theresa Gue, "Triggering Infection: Distribution and Derivative Works under the GNU General Public License," J.L. Tech. & Pol'y, (Vol. 2012), 95 at 113.

²² "MPL 2.0 FAQ," Mozilla, available at <https://www.mozilla.org/en-US/MPL/2.0/FAQ/>.

²³ Heather Meeker, "The Gift that Keeps on Giving – Distribution and Copyleft in Open Source Software Licenses," IFOSS L. Rev., 4(1), pp. 29-40 (2012).

²⁴ "Chapter 18: Special Topics in Compliance," copyleft.org, available at <https://copyleft.org/guide/comprehensive-gpl-guidech19.html> (accessed Jan. 30, 2016).

²⁵ Gue at 101-105.

²⁶ *Id.*

²⁷ 17 U.S.C. §§ 502 and 504.

²⁸ See Kemp at 20.

²⁹ 535 F.3d 1373 (Fed. Cir. 2008).

³⁰ *Id.* at 1381.

³¹ *Id.* at 1382.

³² MDY Indus., LLC v. Blizzard Entm't, Inc., 629 F.3d 928 (9th Cir. 2010).

³³ *Id.* at 941.

³⁴ Victoria Nemiah, "License and Registration, Please: Using Copyright 'Conditions' to Protect Free/Open Source Software," 3 NYU J. Intell. Prop. & Ent. L. (Spring 2014) 358, 384-85.

³⁵ No. A-14-CA-12-SS, 2014 WL 950065 at *4-*5 (W.D. Tex. Mar. 11, 2014).

³⁶ *Id.* (citations omitted)

The NSBA's Top Advice at Your Fingertips: Introducing the NSBA Library

What if you could easily turn to some of Nebraska's best lawyers for their insider tips on the best ways to get something done for a client, or to reduce a tax bill, or save you time and headaches managing the technology in your office? Better yet, what if you could access all the information with a few computer keystrokes? Well, now you can!

The newly-organized online NSBA Library offers NSBA dues-paying members access to the NSBA's vast collection of information on topics ranging from agricultural law to workers' compensation. During the summer, the NSBA staff reviewed and organized thousands of pages of written materials and hours of video to make it easier for practitioners to find useful information. The NSBA Library includes seminar materials and videos, articles from *The Nebraska Lawyer*, forms, practice manuals, inheritance tax worksheets, and more.

The NSBA Library organizes materials by topic, such as Family Law or Real Estate. In addition, lawyers can search the entire NSBA Library using the search function on the NSBA website. For example, using the search terms "adop-

tion" or "deed of trust" or "social media" returns a wide array of information for any of the topics, including articles, seminar materials and videos.

Check out the new NSBA Library on the NSBA's website (www.nebar.com). Under the drop-down menu for "Practice Tools" is a link to the NSBA Library. Select the topic or practice area of interest and see the list of downloadable materials and videos. For some of the most current practice manuals and similar materials, the viewer will be directed to the NSBA Store, where the product can be purchased.

The value of the NSBA Library will continue to grow as new materials are added. The NSBA is interested in hearing from practitioners if there are materials that should be added to or deleted from the Library, are in the wrong practice area, or should be cross-listed in multiple areas.

The newly-organized Library helps the NSBA meet one of its strategic plan goals—"to provide members with valuable, relevant services and resources." Please contact us with any suggestions or comments as to how to improve the NSBA Library or other ways to meet our goal.

