



## 710: Is Open OK? Managing New Open Sources, Resources, Risks, & Rules

**Henry W. (Hank) Jones, III**  
*Attorney*

**Heather D. Rafter**  
*General Counsel*  
Digidesign, a division of Avid Technology, Inc.

**D. C. Toedt**  
*Vice President, General Counsel, & Secretary*  
BindView Corporation

**Mark H. Webbink**  
*Senior Vice President & General Counsel*  
Red Hat, Inc.

## Faculty Biographies

### Henry W. (Hank) Jones, III

Henry W. (Hank) Jones, III is an information technology lawyer and businessperson who operates both a solo law practice and a consulting practice, based in Austin, Texas. His varied roles have included commercial, licensing, and intellectual property law, open source software consultant, intermittent second-chair litigator or expert witness trainer, part-time product manager, and risk manager. He particularly works in software, e-commerce, Internet issues, networks, data, hardware, procurement, print publishing, and other markets. He assists software users, vendors, and intermediaries in open source software counseling, product planning, licensing, risk assessment, policy creation, and training.

Mr. Jones formerly served as both head in-house counsel and a utility senior management infielder for QMS and Ashton-Tate and as vice president., intellectual property development for U.S. Robotics. He also worked in-house at Accenture and Arthur Andersen, handling licensing, e-commerce, Internet issues, outsourcing, alliances, systems integration, new services development, marketing, publishing, training, and other projects, and in private practice in Austin, Atlanta, and Memphis.

Mr. Jones has guest lectured at corporate legal department and management events, and at numerous law, management, engineering, and other industry and academic programs. He has served as chair of ACCA's national Internet IP Issues Subcommittee, chair of the computer law section of the Georgia Bar, chair of the Computer Law Association's annual conference, and guest author for periodicals.

He graduated from Duke University *magna cum laude* and from Vanderbilt Law School.

### Heather D. Rafter

Heather Dembert Rafter is general counsel of Digidesign, a division of Avid Technology, Inc. Ms. Rafter is responsible for managing Digidesign's legal affairs. The company's hardware and software is used worldwide by individuals and companies who create digital audio content, from home music hobbyists to editors of feature films in Hollywood.

Prior to joining Digidesign, Ms. Rafter was an associate at Gibson, Dunn & Crutcher in its San Francisco office and a member of its litigation department. While at Gibson, Dunn & Crutcher, Ms. Rafter represented a variety of media and high technology companies, including Accolade in *Sega Enterprises v. Accolade, Inc.*

Ms. Rafter has been active in the ABA and is currently the chair of the ABA's section of science & technology law. She also is a past chair of the Barristers intellectual property committee of the Bar Association of San Francisco. Ms. Rafter frequently lectures and writes on intellectual property, Internet and computer law issues. She has written articles for various publications, including the *San Francisco Daily Journal*, *Business Law Today*, *Computer Lawyer*, *Loyola of Los Angeles Entertainment Law Review*, *Multimedia Law Report*, *Columbia Journal of Transnational Law*, and *Hastings Communications and Entertainment Law Journal*.

She received her undergraduate degree from Princeton University, *magna cum laude*, and her JD from Columbia Law School, where she was a Harlan Fiske Stone Scholar.

### **D. C. Toedt**

D. C. Toedt III is vice president, general counsel, and secretary of BindView Corporation, a Houston-based software company that provides business policy, IT security, and directory management solutions. Mr. Toedt joined BindView after being outside counsel for the company since its founding.

Mr. Toedt was formerly a partner and member of the policy committee at Arnold, White & Durkee, one of the nation's largest intellectual-property firms (now Howrey Simon Arnold & White). Before law school, he served for five years as a U.S. Navy nuclear engineering officer.

Mr. Toedt is a former member of the council of the ABA's section of intellectual property law, and served as principal drafter of the section's Model Software License Provisions project. He organized and chaired a panel discussion at ACCA's 2001 and 2002 annual meetings on "Ten Things I'm Glad I Knew – or Wish I'd Known – My First Year as General Counsel." Mr. Toedt was the editor and principal contributing author of *The Law and Business of Computer Software*, a one-volume treatise published by West Group. He is active in local community, church, and civic organizations.

Mr. Toedt received both his BA, with high honors and his JD from the University of Texas at Austin, where he served on the editorial board of the *Texas Law Review*.

### **Mark H. Webbink**

Mark H. Webbink joined Red Hat, Inc. as its first general counsel and was subsequently elected secretary and senior vice president of the company.

Prior to joining Red Hat, Mr. Webbink was associated with Moore & Van Allen, PLLC as a part of its intellectual property practice team where his work focused on intellectual property transactions, including software and patent licensing.

Mr. Webbink participates in a number of organizations, including the North Carolina Bar Association, Licensing Executives Society, the board of directors of the Software and Information Industry Association, North Carolina Electronics and Information Technology Association, and International Trademark Association. He has spoken on open source licensing and issues of software patents before the National Academy of Sciences, the Computer Law Institute of the Practising Law Institute, Georgetown's Advanced Computer and Internet Law Institute, the Federal Trade Commission, the U.S. Department of Justice, Congress, and numerous law school seminars.

## Is Open Okay? Managing New Open Source Software Resources, Risks and Rules

American Corporate Counsel Association  
Annual Convention  
“Charting A New Course For You, Your Profession, and Your Corporation”  
Panel 710  
October 9, 2003  
San Francisco Marriott

Henry W. (Hank) Jones, III  
Law Office of Henry W. Jones, III and  
Intersect Technology Consulting

1. What Open Source Software (“OSS”) Is: Broad, Widening, Deepening, Revolutionary, and Part of Important Larger Political/Cultural Trends

What’s new (and contrary to traditional software product licensing)?: source code is given.

What’s new (and contrary to traditional software product licensing)?: the right to modify is given.

What’s new (and contrary to traditional software product licensing)?: the right to redistribute is given.

What’s new (and surprising to many recipients)?: if your client both (a) makes any changes and (b) “distributes” those changes, then the new component(s) must be given to others in source code form (and watch out for what’s defined as “distributing” – getting paid usually isn’t required).

What’s new (and surprising to many recipients)?: often, you can’t choose the terms and conditions on which you redistribute to other parties (rather, you’re required to repeat the same “t’s and c’s”).

What’s new (and surprising to many recipients)?: often, you’re required to either physically ship the t’s and c’s along with the software or specify where the t’s and c’s are available.

What’s old, but still important?: OSS licenses and products/utilities *vary*; the “rules” summarized above do not always apply (a/k/a “the devil’s in the details”); there are several dozen OSS licenses you may encounter (i.e., your clients may invoke) now, with more surfacing.

What's the context?: global Internet access for anyone to anything digital tends to encourage new distribution activities.

What's the business culture?: note that OSS is a "movement" and a "community". OSS advocates and authors usually aren't MBAs, but they're bright and active.

So, the consensus is that there is a new, unprecedented status and trend in the software environment:

" ... The case, regardless of its outcome, also points to a broader issue that will not go away:  
*how to manage the meeting of two worlds of programming. ...*"

*New York Times* 6/14/03 article  
(mainly re. the SCO/Caldera v. IBM lawsuit filed 3/6/03) (italics added)

2. What OSS *Isn't* (i.e., Entirely Unprecedented In Software and Tech. Law):  
Remembering Shareware

OSS isn't the first distribution of software under rules that would shock traditional businesspeople.

For example, programmers and software industry folk have seen (and sometimes used, or even distributed and marketed) "shareware", where an "honor system" was utilized to generate revenue (but source code usually wasn't provided).

3. Some Vendors Don't Necessarily Know All Their *Technology Products' Contents*:  
Industry Traditions, Software Architectures, and *Computer Associates v. Altai*

"/There's something goin' on/And you don't know what is/Do you, Mr. Jones?/"  
Robert Zimmerman, a/k/a Bob Dylan, "Ballad of a Thin Man", from Highway 61 Revisited (1965)

Usually multiple vendors, not a sole creator: Software as received from a vendor frequently includes components obtained from third and even fourth parties (e.g., to avoid "re-inventing the wheel", achieve technical interoperability, support quality control goals, reduce costs, and accelerate time to market).

Passed-through components: A well-accepted tradition in the software industry is "OEM deals" (from "original equipment manufacturer"), where one software vendor licenses manufacturing, distribution, and sometimes modification rights to a licensee for a multi-year duration. So historically it's been rash to assume that a vendor of software or other information technology products delivers a product containing 100% "home grown" software/technology.

Occasional old-era adverse surprises regarding software components: in *Computer Associates v. Altai*, a programmer left one employer (the plaintiff) and joined a competitor (the defendant), but

added the plaintiff's code into his work product for the new, apparently unsuspecting employer, resulting in protracted trial court and appellate litigation in both the U.S. and France.

Now, vendors of information technology products (including peripherals, hardware, and other gear that contain software physically "baked" into "firmware") and related services, can include *free* product components.

Now, vendors may get product components from self-organizing, geographically dispersed, volunteer bands of programmers, without the warranties, indemnification, and commercial context of traditional (i.e., proprietary, for-fee) software component licensing.

4. The Past As Inadequate Predictor: OSS = *New Deliverables, Processes, Authors, and Rules*

Traditionally, licensees/users/customers purchase ongoing technical support (maintenance assistance) from the vendor/creator of the particular software and other information technology product.

Now, software support can be self-help (since users get source code), and/or can seek and receive answer and even fixes (new code) from the "community" (i.e., volunteers responding to questions or requests dispersed via the Internet).

Traditionally, software and other information technology vendors prefer to offer products where only they know or support the product and its components. Such maintenance revenues are long-term and deliver often good margins.

Now, some software and other information technology vendors may choose to compete by reducing their costs (and hence perhaps their product fees/pricing), enabling third party technical support, and reducing development time (accelerating "time to market"), by deploying and disclosing OSS within their products.

Traditionally, software and other information technology vendors can test their products robustly before shipment to customers, since the vendor holds some combination of (a) original source code and original technical design/development documentation, for the portions the vendor wrote, and (b) certain technical materials, warranties, design or integration services, and ongoing support commitments from the "upstream", commercial providers of licensed-in components.

Now, software vendors may prefer to disclaim responsibility for quality-testing of the OSS portions of their product or service offerings, instead referring customers to assurances of source code, "safety in numbers", and/or third-party specialist OSS support service companies.

5. **Traditional Software Procurement and Product Management Methods As Inadequate Protection:** OSS = New Skills and Processes For Managers and Their Lawyers To Consider, Learn, Test, Invest In, Implement, Document, and Refine

Traditionally, customers and their lawyers have sought and usually received certain carefully crafted warranties of substantial software quality.

Now, software providers may respond, “we didn’t write that and you received it free, so, no, but it’s OK anyway.”

In recent years, customers and their lawyers sought and usually received warranties that the software doesn’t contain viruses, Trojan horses, worms, and other “harmful code”.

Now, software providers may not offer such assurances. And providers may be a band of individuals, not a commercial vendor with (hopefully) assets and even insurance coverage.

Traditionally, software vendors haven’t stressed prohibiting programmers from breaking corporate rules (it’s been a few sales folks who were believed to sometimes merit monitoring).

Now, with reduced staffing, reduced sales revenues, increased “time to market” pressures”, there may be motivation for programmers to “grab off the ‘net” software product components, without prior consultation with management or counsel (again, the occasional rogue salesperson analogy).

6. **New Sources of Foreseeable Risks:** Why OSS = New Process, Liability, Operational, and Embarrassment Challenges

A. Dynamic Organizational Boundaries of Your Employer

Programmers and software development processes are less easily managed, as they are increasingly dispersed, using “telecommuting”, after-hours work at home, “offshore-sourcing” to (usually cheaper) programmers in other countries, and round-the-clock schedules.

Containment and buffers aren’t realistic: every programmer has Internet access to free sources of OSS (and probably should, to enable their ongoing skill development).

B. Unauthorized OSS Deployment By Individual Employee(s) and Contractors

Pressured programmers may choose the “short-cut” of “baking in” OSS – especially if there has been no corporate policy, education, or enforcement regarding when to not use OSS (or to ask for clearance).

Headline news coverage regarding the OSS trend makes OSS “cool” (e.g., see March 2003 cover stories in both *Business Week* and *CIO* magazines).

### C. Uncertain Ownership and Genealogy of OSS Code

OSS development groups are usually self-organizing bands of volunteers: did individual programmers get corporate/employer clearance to participate?

OSS development groups are usually self-organizing bands of volunteers: were intellectual property waivers and/or assignments executed?

Have proprietary code components percolated or leaked into OSS offerings (as asserted in *SCO v. IBM*), despite the “code review”, “version control”, and “review team” processes and filters maintained by many OSS groups?

“... it’s not uncommon for various Linux strains to borrow a program from here, co-opt some code from there, and so on. Most Linux coders wouldn’t knowingly build a copyrighted program into their code, but some are unaware -- or choose to be unaware -- of a code’s legal lineage.”

From “A Big Test for Linux”, Eric Hellweg, *Business 2.0* magazine, 1/27/03 (electronic magazine [“e-zine”] version)

Intellectual property infringement assertions against some iterations of freeware isn’t new. Compare the 1992-1994 litigation among vendors of the Unix operating system software.

### D. Uncertain Supply Lines

Some software development organization (both in-house and vendor teams) lack adequate, granular, enforced “version control”, “configuration management”, record-keeping, and other systems and tools.

Who did quality-checking against viruses and “harmful code”? How well?

### E. New Terms and Conditions in New Licenses Not Previously Interpreted by Either Courts or “Industry Tradition”

There exists no known court ruling adjudicating the enforceability of any OSS license. (*Progress Software v. MySQL*, addressed in an attachment, settled before a definitive ruling.)

Many OSS licenses have been drafted by programmers and businesspeople, not lawyers.



Software industry veterans have not reached consensus regarding the interpretation of some of the key provisions in the best-known OSS license, i.e., the “General Public License” (a/k/a “the GPL”) version 2.0 as promulgated by the Free Software Foundation (e.g., when is a “derivative work” created, depending on the particular architecture/design, content, packaging, and shipping of modified or supplemented software).

Emerging, new software technologies, practices, concepts, and vocabulary may challenge the clarity and adequacy of pre-existing OSS license texts (e.g., regarding the Java programming language?).

#### F. Disagreements and Politics Among OSS Advocacy Groups and Leaders

Programmers frequently are bright and independent-minded (e.g., see [www slashdot.org](http://www slashdot.org), the important Web site “portal” for “news for geeks” and related chat and debate).

Leadership of the “OSS movement” is dispersed, among the older Free Software Foundation, Open Source Initiative, relatively recent for-profit OSS services and software vendors, and other organizations and individuals.

#### G. Increasing Desire and Ability by OSS Advocacy Groups to Enforce OSS Licenses

The pro bono General Counsel of the Free Software Foundations claims to have challenged and remedied dozens of OSS breaches by various parties.

A leading vendor of OSS, MySQL, donated \$25,000 to fund OSS forensics activities by the Free Software Foundation.

Self-appointed OSS enforcers have surfaced in the “techie” community and globally announced what they believe to be discoveries of “OSS cheating” by major corporations (e.g., on the influential [www slashdot.org](http://www slashdot.org) portal).

Compare the “whistle-blower” trends in other environments.

Compare the frequent triggering of software piracy audits and enforcement by “disgruntled insiders”.

### 7. Evidence That New OSS Risks Are *Not Adequately Managed Today*

#### A. *Progress Software and NuSphere v. MySQL*: Publicly Traded U.S. Software Vendor Screws Up regarding OSS Compliance

See attached article (originally published in 8/02 issue of *Linux Journal*).

- B. Sigma Designs: Publicly Traded U.S. Technology/Peripherals Vendor Gets Caught By OSS Group

See attached two press/Web releases.

- C. *Blizzard v. BNETD* (i.e., Intellectual Property Litigation regarding “Battle.net” Game Extension): “Posse” of Dispersed Programmers Gets Sued By Proprietary Vendor

See pleadings collected at [http://www.eff.org/IP/Emulation/Blizzard\\_v\\_bnetd/](http://www.eff.org/IP/Emulation/Blizzard_v_bnetd/).

- D. “You Don’t Know What You Don’t Know”

Some software vendors presently lack any adequate, supervised, disclosed, or documented policy or processes for deciding on or documenting the inclusion of OSS.

Many internal-use software development teams (i.e., inside companies that don’t license or sell the resulting software) presently lack any policy or processes for deciding on or documenting the inclusion of OSS.

8. Porn Sites, Harassing Email, Sexual Harassment, and Environmental Protection: Isn’t OSS A New Logical Candidate for Creation and Implementation of Corporate Compliance Policies

Isn’t one value-added role of counsel the identification of new corporate business risks, and then helping to organize and deliver processes and tools to manage and mitigate those risks?

Does the new OSS phenomenon have the magnitude of risks as other major corporate risk reduction trends of recent decades, even though (presumably) there (usually) is no legislative or regulatory mandate to act in this particular domain? (This assumes that your employer’s data processing operations aren’t subject to special industry-specific or other regulatory requirements.)

Compare the other, recent trends of creating and implementing corporate policies regarding appropriate use of email, appropriate third party Web sites to visit using company time and equipment, safe telecommuting (i.e., home computing to do employer tasks), and the like.

Will in-house counsel be proactive and adopt the additional roles of part-time educators, “process engineers”, and risk managers?

9. Action Recommendations

Be flexible: “Just say no” isn’t perceived as a value-added response, when a new technology offers reduced up-front costs, possibly (depending significantly on the particular details and

circumstances) new functionality, possibly better quality (though traditional vendors would argue against this and other alleged benefits, often very convincingly), possibly a broadened choice of suppliers (though arguably OSS groups are not “vendors”), possibly benefits in recruiting and retaining smart information technology personnel, and perhaps other advantages.

Determine and define specific new corporate rules and processes to address a new phenomenon: Develop, announce, implement, and enforce a corporate policy specifying both (a) generally, when, where, how, why, and by whom OSS may and may not be used, and (b) who to contact and who makes decisions in new, uncertain, or challenged situations.

Verify?: Consider auditing your software and information technology infrastructure. “Ignorance of The Law Is No Excuse.” Other corporate goals that can be addressed in the same initiative include (a) avoiding unrecognized infringement of third-party traditional, proprietary software, (b) assessing the adequacy of security safeguards (e.g., for networks, data, applications, and remote access), and (c) exploring opportunities for cost reductions (e.g., by consolidating and/or updating software applications, possibly outsourcing to specialist vendors, and possibly deploying OSS).

Remember and protect the Attorney-Client Privilege: You may not like, or want disclosed, what an initial OSS audit or investigation reveals. Do you want a lawyer who merits and can maintain “the privilege” to manage or at least coordinate the effort?

“Check your tires’ treads”: Review and update your in-licensing and procurement contracts for incoming information technology. Do they address OSS at all? If so, how well? Have you assessed modifying your testing, technical support, training, warranty, indemnification, and other contract provisions? Do they require disclosure of past, present, or planned future OSS utilization by products and services vendors? With what degree of granularity? With what degree of effort/diligence and certainty? Do they cover indirect (e.g., third and fourth party) sub-vendors? What does your employer do with this information? Is the resulting OSS usage data updated periodically?

Check your digital, virtual “loading dock”: Are any of your clients now distributing OSS to customers, suppliers, or other “business partners”, even fee-free? For example, has OSS been emailed out or offered in the creation or operation of a corporate “extranet”? Has your employer complied with each of the applicable OSS licenses triggered by the particular utilized code? Who has determined which OSS went “inside”? Are such initiatives planned or likely for the future?

Inspect how the digital “spaghetti” or “sausage” gets made (or arrange for somebody to do such “health” inspection): Usually lawyers don’t know about or get involved regarding the processes of software design and development by their fellow employees (unless software is a product of or otherwise strategic to the company). Now it’s time to provide guidance and parameters to internal software developers, even if the software is currently intended only for “internal use”,

since later external distribution could trigger a variety of new legal OSS (and other) obligations and risks (e.g., see above regarding “extranet” and “supply chain” software sharing).

Preparation beats litigation, perspiration, or even inspiration: Offer (and even require attendance at) OSS-specific training for relevant employee groups. “Preventive law” is a good and time-honored investment in other segments of information technology; OSS risk management (and probably approved enablement and utilization) should be added to the scope of skills building. Plus, such sessions are great “listening posts” both (a) to get the real story regarding corporate software activities and plans, and (b) to enlist support from law and compliance oriented employees in other departments. (“Compliance paralegal” within M.I.S. or R&D, without a charge to the Legal Department budget, anyone?)

Do the “Texas Two-Step” (or, “Require Blood Tests Before Marriages”): Conduct robust due diligence before signing or closing acquisition, joint venture, “alliance”, investment, and other serious transactions. Savvy technology companies often require a “code review” before obligating themselves to complete a deal with another company if software is a significant portion of the other side’s assets, strategy, operating infrastructure, or intellectual property. Some deals are cancelled, when unauthorized or unexpected code is found. Don’t be surprised if your company’s intended “partner” has OSS problems inside.

Get “hep”: Has “proprietary” become “un-cool”? Recognize that the modern business world offers new variations on intellectual property options that most of us never learned in law school. (E.g., visit [www.creativecommons.org](http://www.creativecommons.org).)

Become aware that your clients may be wrong-headed regarding third party intellectual property: Get in synch with the new global digital real world. Recognize that your clients may hold relatively recent, arguably wacky (at least from a traditional lawyer’s perspective) ideas regarding procuring digital tools and technology. Since everyone’s clients can download and deploy “freeware”, the always-on, accessible-to-everyone nature of the Internet presents challenges to all lawyers, not just to lawyers for vendors of music, films, proprietary software, and other copyright-based industries.

Meet your geeks: Do you know their policies? Do they know what their staff, independent contractors, and vendors are doing? With certainty? Accurately? Why so?

Communicate creatively: OSS is new and unfamiliar, to many clients and nearly all lawyers (both in-house and out-house). Use metaphors to both explore and explain the components, processes, risks, and decisions of this new legal territory. Real estate development (with its multiple parties, long-term considerations, occasional disappointments, and resulting litigation) is one useful source of analogies. Medical phenomena and safe operation of vehicles may be others. For example, is belated discovery of unauthorized OSS inside a supposedly traditional, proprietary software application the business equivalent of a tumor diagnosis? (Benign or malignant? Early stage or advanced? Lots of scary uncertainty, either way.) Clients don’t

appreciate discussion of precedent and *res judicata*, but do understand a car wreck or emergency room visit when they see or hear one.

“Constant vigilance is the cost of freedom”: Review and update your company’s OSS policies, practices, and awareness periodically – e.g., every 6 months. New licenses, applications, business practices, (eventually) court rulings, and maybe (eventually) regulations likely will trigger needed updates and enhancements to this new, young part of your corporate business processes.

Be multi-disciplinary: Try to staff your OSS adaptation team or project with representatives from not just Legal, but also M.I.S./I.T., Procurement, Product Strategy, H.R., and possibly other corporation functions. Compare the 6/14/03 *New York Times* article regarding the OSS issues team inside vendor Hewlett-Packard.

Consider “upstream” obligations: When making your OSS decisions, determine whether prior contractual commitments (e.g., to investors, suppliers, customers, or other parties) or common law obligations may impact how you act regarding OSS. Your own views regarding costs, quality, reliability, security, intellectual property, innovation, staffing, and other issues may not be the only input.

Hand off the risk?: Consider exploring insurance coverage, for special, high-risk scenarios. Carriers can innovate in respond to new market demands, and the Lloyds/London custom (“manuscript”) insurance market might work.

10. **Expanding Relevance**: Precedent for Upcoming Challenges in Other, Non-Software Digital Environments: Predicted “Open Source” Impacts In Bioinformatics, Publishing, Music, and Other Technologies, Tools, and Industries

Be aware that the OSS “movement” and “community” is causing and supporting similar experiments and advocacy in non-software domains.

Anything that can be digitized can be distributed over the Internet, private phone lines, or even through the air (via wireless transfer). Pharmaceutical, data base, entertainment, and other lawyers need to know about, watch for, and act to help manage “open source” activities by their clients and in their industry.

11. **Homework Resource Suggestions**

Best overview: *The Cathedral and The Bazaar*, Eric Raymond (available both on his Web site and in traditional hard-copy book form) (advocacy, explanation, and anthropological analysis of OSS by an active veteran of the OSS world).

Good more in-depth history, though somewhat outdated: *Free For All: How Linux and The Free Software Movement Undercut The High Tech Titans*, by Peter Wayner.

S.E.C. filings by publicly-traded OSS-specific software/services vendors.

Go meet and talk separately with several OSS geeks (whose views and experience may vary).

Rent or buy the video or DVD of the documentary film "Revolution OS" (covering the OSS "movement").

Other.

Attachments:

"XVID Team Requests Sigma Designs' [sic] To Halt Copyright Infringement" (global press release by ad hoc group of programmers asserting unauthorized use of their OSS code by publicly traded U.S. based technology vendor, and attaching engineering analysis as evidence of breach of OSS licensing rules)

"Mea culpa" press release (Web site posting) by Sigma Designs apologizing for the OSS misuse

"How a Poor Contract Sunk an Open-Source Deal", *Linux Journal*, August 2002, by the speaker, also available at <http://www.linuxjournal.com/article.php?sid=6025>

Sample risk disclosure regarding OSS incorporation by a publicly traded traditional/proprietary software vendor, from its recent Form 10-K

Sample prohibitions on OSS use, extracted from recent merger/acquisitions agreements

Sample prohibition on OSS use, extracted from October 2000 software licensing-out agreement (without mentioning ever "open source" or any particular OSS license)

This handout is intended to be supplemented by the speaker's live comments, should not be attributed to any future, current, or past client or any other third party, and represents only a portion of the speaker's current personal opinions on this evolving topic.

© 2003 Henry W. Jones, III

## XVID TEAM REQUESTS SIGMA DESIGNS' TO HALT COPYRIGHT INFRINGEMENT

22nd August 2002

**ERLANGEN, GERMANY – August 22nd, 2002 – The XVID development team, author of the popular XVID MPEG-4 video codec, claims that Sigma Designs' REALmagic MPEG-4 Video Codec is an illegal copy of the XVID software and publicly requests the company to stop violating their software license and copyrights.**

XVID is a leading open source MPEG-4 video research project: software distributed by XVID is covered by a Free Software license, the GNU General Public License (GNU GPL). The XVID team announced that Sigma Designs' REALmagic MPEG-4 Video Codec includes wide portions of XVID codec software. By not offering a corresponding source code distribution and by claiming sole authorship on the product, Sigma Designs' Inc. is violating the GNU General Public License and the copyrights of the XVID authors.

XVID learned about the license violation in early July, soon after the initial release of the REALmagic software (version 1.0). Sigma Designs' were immediately contacted, and replied confirming the violation and promising to replace all violating code.

Version 1.1 of the REALmagic software was released on the 9th of August. After examining the new version, XVID developers concluded that the violating code was not replaced, but disguised by programming and compiling tricks. Sigma Designs' were again contacted and asked to remove the REALmagic download link from their website. Thus far, they have not shown any sign of cooperation.

In a statement to the XVID development team, project founder Michael Militzer showed his disappointment regarding Sigma Designs' behaviour: *"We have been quite reasonable and have given Sigma Designs' ample opportunity to resolve this issue. Apparently none of our demands have been taken seriously. Nearly two months after the initial release of the REALmagic MPEG-4 Video Codec, Sigma Designs' is still knowingly infringing the GNU General Public License."*

Militzer believes this infringement might be of high general interest: *"This is an unfortunate event, not only for us but for the whole Free Software movement. Therefore we hope to receive wide support from the Free Software community in our efforts to convince Sigma Designs' to respect the terms of the GPL."*

Evidence supporting the claim has been published on the XVID website.

- [http://www.xvid.org/v1\\_0\\_comparison.pdf](http://www.xvid.org/v1_0_comparison.pdf)
- [http://www.xvid.org/v1\\_1\\_comparison.pdf](http://www.xvid.org/v1_1_comparison.pdf)

### About XVID<sup>1</sup>

XVID is a leading open source MPEG-4 video research project, founded by the German student Michael Militzer in August 2001 to continue the efforts of DivXNetworks' former OpenDivX project. Today, the XVID project consists of users and developers from all over the world. XVID publishes all its software under the GNU General Public License (GNU GPL).

### About Sigma Designs Inc.<sup>2</sup>

Sigma Designs' headquarters are located in Milpitas, California. The company specializes in MPEG based video hardware for encoding and decoding. Recently Sigma Designs' introduced the Xcard, the first consumer hardware MPEG-4 decoder in the form of a personal computer add-on card.

### About GNU GPL<sup>3</sup>

The GNU General Public License is the most frequently used software license for Free Software development and is supported by the Free Software Foundation (FSF). Software distributed under the GNU GPL grants everyone modification and redistribution rights, on the condition that derived or redistributed software carries the same license.

###

### Contacts

For contacting the XVID team please use the e-mail addresses:

[contact@xvid.org](mailto:contact@xvid.org) or [contact@xvid.de](mailto:contact@xvid.de).

- **Daniel Smith** (USA and Canada)
- **Michael Militzer** (Germany and international)
- **Christoph Lampert** (Germany and international)
- **Edouard Gomez** (France)

---

<sup>1</sup><http://www.xvid.org/>

<sup>2</sup><http://www.sigmadesigns.com/>

<sup>3</sup><http://www.gnu.org/copyleft/gpl.html>




[company](#)
[investors](#)
[products](#)
[partners](#)
[support](#)
[sales](#)
[careers](#)
[home](#)

## REALmagic® MPEG-4 Video Codec

### Software Encoder and Decoder for ISO MPEG-4 Video

Interested Parties concerning Sigma's MPEG-4 CODEC

Thank you for contacting Sigma Designs and requesting information that relates to our MPEG-4 CODEC and the availability of its source code. We would like to take this opportunity to address the relevant issues that have been raised.



To begin with, Sigma developed an MPEG-4 CODEC to assist in the proliferation of MPEG-4 content and to ensure that users can create content libraries compatible with the ISO MPEG-4 video specifications and its implementation in silicon. Fulfilling this goal was carried out in two steps. The first was the introduction of an MPEG-4 CODEC, provided free of charge, so that worldwide users could begin encoding new content. The second was the release of source code, so that the development community could continue with technical improvements. Sigma never intended in making, nor realized, any profits from this code.

Several weeks after the CODEC was first released, Sigma was contacted by the XVID development team regarding the use of certain portions of their code. Upon examination, it was determined that one of our programmers, unbeknownst to management and contrary to Sigma's policy, had utilized some routines posted by XVID as open source. During the past four weeks, Sigma had communicated with XVID to resolve the situation. As a result, Sigma has decided to make the current version of the MPEG-4 CODEC available under the GPL license.

Sigma is a supporter of the Linux operating system, appreciates the work being done by the open source community, and continues to issue certain other code under open source arrangements. Though we believe that we have acted as expediently as possible, Sigma Designs sincerely apologizes to the open source community for this inadvertent use of GPL code and for the several weeks it took to resolve the situation.

Sincerely,  
Ken Lowe

[http://www.sigmadesigns.com/products/RMP4\\_video\\_codec.htm](http://www.sigmadesigns.com/products/RMP4_video_codec.htm)



## **How a Poor Contract Sunk an Open-Source Deal**

**Date:** Thursday, August 01, 2002

**Topic:** Linux in Business

Henry W. Jones, III

Why the Progress and NuSphere vs. MySQL AB litigation is about sloppy deal making, not open-source integrity.

# **How a Poor Contract Sunk an Open-Source Deal**

*Why the Progress and NuSphere vs. MySQL AB litigation is about sloppy deal making, not open-source integrity.*

*by Henry W. Jones, III*

Many describe a new continuing lawsuit in federal court in Boston as ``The first litigation testing the validity and enforceability of the General Public License" (GPL). So what?

Will this litigation really impact the future of Linux programmers?

Does this dispute matter for companies betting their business models on the open-source trend? Will the judge get the chance to punish an arrogant American software vendor that broke the long-known rules of GNU and thereby defend the OSS cause, as some OSS advocates have suggested?

Sorry, probably not. Yes, the case is important. Yes, it is apparently the first GPL court test, by consensus. But it won't foretell the OSS future because it's a dispute about an extraordinarily poor contract in a context of chaotic, changing communications between the parties.

You can't project the prospects of a programming language from analysis of one short, poorly documented application coded in that language. And in this case, the underlying contract is an outlier that's so far from norms of modern prudent software management and licensing practices that by many orders of magnitude, it's off the map. It ultimately will prove more relevant for "Software Product Management 101" and "Beginner Software Contracts" training than for refining OSS strategies.

## **Snapshot of a Train Wreck**

The story is told in the publicly available court pleadings. The contract underlying the litigants' dispute is a disclosed attachment to the answer filed by the Swedish authors of the well-known MySQL OSS database to the lawsuit initiated by the US software

publisher/remarketer. (So the contract and the parties' various arguments, e-mails and affidavits are ``open source" for tech managers, lawyers and trainers to study and use to improve work processes.)

This author obtained from court pleadings the original international agreement by which a publicly traded, long-established business software company based in Massachusetts obtained remarketing rights from a young, offshore, small developer in Sweden. Ugly surprise: these two companies agreed to do a big-impact, large-dollar deal on a mere nine-paragraph contract. The agreement ran all of 1.25 pages.

Progress Software agreed to pay roughly \$300,000 US to a dynamic foreign company in a new, unfamiliar (to Progress) industry segment, on the equivalent of the proverbial envelope. MySQL AB, the Swedish company, blessed the Massachusetts vendor's procurement of its key product by a short statement indicating some future contract would be utilized ``later", triggering ``a total of up to \$2.5 million". The resulting fight shows precisely why experienced business people (including lawyers) frown at the optimistic idea of ``let's just trust each other and figure out later the deal and the details."

What's wrong with a little brevity and trust? Think of it this way: why do surgery before taking x-rays or reviewing a medical history? Why not dive head-first in to an unfamiliar river? You can both get hurt

and hurt others by launching a major software initiative--OSS or proprietary--without first figuring out the basic rules. That's what happened here.

One purpose of most contracts is similar to the norms of much data processing: benchmarking, testing and standards. Here, fragmentary code got shipped. That is, an incomplete ``agreement" was relied upon for too much action, too soon.

## **Deafening, Deadly Silence**

What did this short and ultimately bitter contract omit? The majority of terms and conditions found in most software agreements, that's what. Conspicuous by their absence, among other points, were 1) When would the expected ``later, superseding agreement" be completed? 2) Within what parameters for the business terms? 3) Exactly what degree of service would be required and provided for technical support? What did they mean by ``enterprise level support" and ``existing electronic support channels"? 4) Who would be the designated liaisons for intercompany coordination? 5) What does it mean to give your licensee ``fair use" rights to your key trademark, as MySQL AB blessed here? What particular variations would be permitted and excluded? 6) What ongoing product enhancement services by the original author would be assured? 7) How would disputes be resolved or arbitrated, if necessary? 8) If there's a dispute due to one party's fault, will the nonbreaching party get its

enforcement costs and damages reimbursed by the defaulting party?  
9) Why omit all the often-derided generic or "boilerplate" provisions that are included in most contracts precisely because they help prevent disputes and enable enforcement?

## **Learning Lessons from Others' Wrecks: Code Your Contracts Like Your Software**

Most modern, mature software businesses recognize the many issues that can and do arise in a software distribution deal. They design their deal (e.g., in a "terms sheet" or outline), then "code" (i.e., write a draft contract), then test and document their *agreements* (i.e., negotiate and refine the base contract and write and revise the necessary exhibits), just as they do their *applications*.

For example, many software projects identify "user requirements" in detail and in advance. This deal apparently lacked a joint "terms sheet" or "deal summary memo" as the anchor for the agreement.

Most applications get a look-over for quality control by programmer colleagues. Automated code-testing tools get deployed in some complex environments. This contract presumably was shipped out as the handiwork of one individual, or at least of a very small team.

Savvy software professionals include error-message features. This oblique agreement lacked the typical "notice of breach, then opportunity to cure the breach" provision.

Experienced coders include header files and other technical documentation in their work to assist later revisions and debugging. In your software transactions, include specified modes of communications between the author and publisher companies. Decide up front which particular individuals have the authorization to pass commercial instructions, objections and suggestions to some specified person(s) in the other organization.

### **Frightful Images: Ships Passing in the Night**

The contract's brevity means the parties may raise legal issues that will muddy the waters or at least defer the outcome. Remember, the wheels of the justice system can grind very slowly, at least in the US.

OSS loyalists hoping for court affirmation of the GNU model may be frustrated: both sides of the suit have already raised legal arguments unrelated to the OSS issue. For example, MySQL AB has already obtained (on February 28) a partial injunction against Progress and its young OSS subsidiary NuSphere, but on trademark law grounds, not enforcement of the GPL. The federal judge found the GPL issue too uncertain to adjudicate in this litigation's early, summary phase.

Then there's the legal doctrine of "mutual mistake". A contract sometimes can go unenforced when both parties inadvertently hold different, though reasonable, interpretations of the deal's predicate and terms. The classic case involves a similar cross-border mishap.

## When Going to Rome, Study Ahead

The rashness of this saga is underscored by its multicountry context. Transnational transactions merit extra thinking and terms, just like multinational applications often require more modular screen messaging, two-byte code (for Asian character sets), accommodating different operating system iterations and other shrewd coding.

Doing deals with foreign companies requires extra consideration. For example, many offshore companies prefer (or insist on) the use of arbitration to resolve disputes, both as part of a strong cultural tradition and to avoid the rumored American tendency toward premature, extended and expensive litigation. (Here, the litigants filed 73 different court pleadings in the initial nine months of the case, with no end in sight.)

World travelers arrange translators, confirm supply lines and determine local communication protocols *before* setting out. In international contracts, many companies take similar extra steps. They pre-agree on minimum collaborative product planning, contractually commit to visit each other's headquarters and meet at major global tradeshow and include other contractual "glue code" to help refine the relationship. Common sense says to develop a map when venturing into unfamiliar territory. Here, the parties got lost and found themselves in court, with the resulting marketing disasters, big litigation bills and an uncertain product road map.



## What to Think; What to Do

Some in the OSS community have attacked Progress and NuSphere, citing the accurate but fragmentary story that the MySQL code got modified and then marketed via a proprietary license, not the GPL or some other OSS license. True, NuSphere modified its model to use GPL, and in NuSphere's view thus fixed a mere short-term oversight. But that's not the full story. The pleadings suggest another perspective: criticize Progress instead for letting some product manager do a poorly documented contract, presumably without coordinating with counsel and other colleagues. Sentence this individual to attend a licensing workshop. Maybe commute the sentence due to time-to-market competitive pressures. And then bet good money that next time both companies will use traditional, coherent, complete software contracts, after learning from spending big bucks on litigators and losing time, managerial energy and market goodwill.

The Progress-NuSphere-MySQL fight ultimately may prove to be just another chapter in the long book of companies who practiced ``ready, fire" without adequate ``aim".

---

UNITED STATES DISTRICT COURT  
DISTRICT OF MASSACHUSETTS

PROGRESS SOFTWARE, CORP., et al.,	)	
	)	
	)	
Plaintiffs,	)	01-CV-11031 (PBS)
	)	
v.	)	
	)	
MySQL AB, et al.,	)	
	)	
	)	
Defendants.	)	<b>DECLARATION OF EBEN MOGLEN IN SUPPORT OF DEFENDANT'S MOTION FOR A PRELIMINARY INJUNCTION ON ITS COUNTERCLAIMS</b>

EBEN MOGLEN, ESQ., under penalty of perjury, deposes and says:

1. I am over eighteen years of age and am competent to testify as to the matters here set forth. I make this affidavit on the basis of my personal knowledge.
2. I am Professor of Law at Columbia University Law School, where I have taught since 1987. I have been a member of the Bar of the State of New York since 1988.
3. Before joining the Columbia faculty, I was law clerk to Judge Edward Weinfeld of the United States District Court for the Southern District of New York, and to Justice Thurgood Marshall of the United States Supreme Court.
4. In 1985 I graduated from Yale Law School and simultaneously completed resident work on my Ph.D. in History, which was awarded, with distinction, on completion of my doctoral dissertation in American legal history, in 1993.
5. From the age of thirteen until I began my judicial clerkships I was employed, part-time and full-time throughout my educational career, as a professional computer programmer. From 1979 to 1984 I was employed by the International Business Machines Company as a designer and implementer of advanced computer programming languages. I consider myself an expert in the design of programming language systems and utilities.
6. My academic research and writing concentrates on the legal changes brought about by digital com-

puters, viewing those changes in technological and historical perspective. I teach courses entitled "Law in the Internet Society," "Computers, Privacy, and the Constitution," and "Perspectives in Modern Legal Thought."

7. I have published widely in these fields, and in the field of American legal history, specializing in the development of law in English-speaking North America from the inception of colonization through the American Revolution. A copy of my *curriculum vitae*, including a list of my publications, is attached hereto as Exhibit A.

8. Since 1994 I have served *pro bono publico* as General Counsel of the Free Software Foundation. I have been a member of the Foundation's Board of Directors since 1999.

### **The Free Software Foundation**

9. The Free Software Foundation ("FSF") was founded in 1985 by Richard M. Stallman, who remains its President. The Foundation is a 501(c)(3) organization incorporated under the law of the Commonwealth of Massachusetts, with its primary place of business in Boston, Mass.

10. The FSF's mission is to encourage the creation and distribution of computer programs, technical documentation, and other related materials that can be freely copied, modified and redistributed by their users. FSF refers to such computer programs as "free software," where the word "free" refers to freedom, not to price. FSF believes that by giving all users the right to copy, modify, and redistribute software, the ethical obligation to maintain freedom of thought is honored. Software that can be freely shared is also of inherently higher quality, because everyone who uses the software can experiment with improvements, and can fix mistakes that are discovered. Because anyone who makes such fixes is allowed to distribute those fixes or improvements, the quality of the software increases exponentially over time, and all programs can be reused for new purposes.

11. FSF supports the development and distribution of free software in two basic ways: it writes and distributes free software of its own, and it helps others to write and distribute such software, primarily through the development and publication of the copyright licenses and associated legal materials that facilitate this hitherto-unusual means of software production.

12. Before the creation of the Foundation, Mr Stallman had begun designing and implementing an entire free software operating environment usable on all computers from the most basic facilities to the most advanced applications. He called that planned system "GNU." Over the course of the 1980s many components of GNU were written and tested, some by Mr Stallman himself and many others by programmers who contributed to the projects and assigned their copyrights in their portions of the programs to the Free Software Foundation.

13. In the early 1990s a young Finnish programmer named Linus Torvalds began to assemble the innermost portion of a computer operating system, known in the trade as the "kernel," first as a personal learning exercise and then as a cooperative project over the Internet, ultimately involving thousands of other volunteers. Mr Torvalds called his kernel "Linux," and he designed it to work compatibly with the other parts of the GNU system designed and incrementally implemented by Mr Stallman and FSF. The result was a system that combined Mr Torvalds' Linux kernel with the other GNU components to make a system that FSF calls GNU/Linux, and which is widely but misleadingly called "Linux" in general parlance.

14. Throughout the 1990s the GNU/Linux operating system became explosively popular with technically-sophisticated users and businesses around the world. The production model of free modification and redistribution permitted rapid development of thousands of applications for the system, and its compatibility with the Unix operating system originally designed and implemented by AT&T made possible the ready adaptation or "porting" of most applications designed for Unix, itself very widely used in technical, academic, and scientific environments. It has been widely reported that GNU/Linux is now the fastest growing operating system in the world for "server computers," which are those computers that perform the tasks required by large networks: file sharing, World Wide Web publication, etc. GNU/Linux is now also used in the smallest computers in the world, including "personal digital assistants" or "palmtop" computers. Because, in compliance with the terms and conditions of the GPL, anyone may freely copy any or all of the programs contained in the system, centralized sales and use figures do not exist, but even a conservative estimate of the number of computers using the operating system and associated application programs throughout the world would reach

into many tens of millions.

15. Although high technical quality and reliability is certainly partly responsible for the success of GNU/Linux, the legal institutions that facilitate this apparently counter-intuitive phenomenon of large-scale non-hierarchical production successfully competing against global corporations are of even greater importance. These institutions depend on a simple but far-reaching employment of copyright law through the GNU General Public License, under which much free software is distributed.

### **The GNU General Public License**

16. The GNU General Public License ("GPL") is the legal heart of the free software movement. The goal of the GPL is to use copyright law to create a "commons," a collection of shared resources to which anyone can add, and from which anyone can borrow freely, but from which nothing can be permanently removed. This concept, of using copyright to create a commons, rather than a domain of exclusive ownership, is sometimes called "copyleft," and the GPL is an example of one form of "copyleft license." A copy of the current version 2 of the GNU GPL, first published in 1991, is attached hereto as Exhibit B.

17. Free software is not in general in the public domain. If contributors to free software projects were placing their code in the public domain, it could be immediately incorporated by others into proprietary, non-free projects, from which those appropriators could derive value without returning anything to the commons. Instead, free software is copyrighted, but through the terms of the GPL, copyright is used to protect the common interest without excluding anyone from the rights to execute, copy, modify, and redistribute that are of the essence to free software.

18. The GPL is a very simple form of copyright license, as compared to other current standards in the software industry, because it involves no contractual obligations. Most software licenses begin with the exclusive rights conveyed to authors under copyright law, and then allow others access to the copyrighted work only under additional contractual conditions. The GPL, on the other hand, actually *subtracts* from the author's usual exclusive rights under copyright law, through the granting of unilateral permissions. When a work of copyrighted software is released under the GPL, all persons everywhere observing its terms are unilaterally

permitted all rights to use, copy, and modify the software. Because these permissions are unilaterally given, users who wish only to use the software themselves, making copies for their own use, or who wish only to make derivative works for their own use, do not have to “accept” the license, because they have no reciprocal obligations under it.

19. If a user wishes to redistribute software she has received under the GPL, whether in modified or unmodified form, the license permits that activity as well. Here, however, the permission is qualified by three primary conditions:

- Redistribution must itself occur under GPL and only GPL, with no additional license conditions. (See Exhibit B, §2(b));
- Redistribution must include “source code,” the human-readable form of computer programs that allows programmers to understand and modify computer programs for themselves, as opposed to “object code,” which is the “machine language” version of computer programs that is very difficult for programmers to understand or modify. (See Exhibit B, §3(a)); and
- Redistribution must include a copy of the GPL, so that users are aware of their rights to use, copy, modify and distribute, and so that anyone engaged in redistribution is also aware of the conditions under which redistribution is permitted. (See Exhibit B, §1).

20. As a result of these conditions on redistribution, the GPL achieves the goal of creating a commons. Anyone can copy and modify program code released under the GPL, but no one can combine that program code with any other code and then release the combination on non-GPL terms. Anyone who contributes program code to a GPL-based programming project knows that her contribution will remain freely available for others to use, fix and improve, but that no one will be able to exclude others from having the same rights. The GPL uses copyright doctrine to achieve the result of the principle that we should all “share and share alike.” Thus anyone who combines GPL-licensed software with other program code must release the combined work under GPL, and must provide the source code for the entire derivative work.

21. Because anyone in possession of a program released under GPL must be in actual possession of the license itself, the licensor is entitled to presume that anyone engaged in redistribution is actually on notice of the only terms on which redistribution is permitted. Redistribution on any other terms is intentional violation of the GPL. (See Exhibit B, §5).

22. The GPL is specifically designed to be a license for decentralized distribution, in which everyone can share programs and improvements with anyone else. This means that program code can cross national borders and otherwise propagate in uncontrolled ways. For this reason, the GPL makes special provision for dealing with the consequences of license violation. Under §4, any licensee who violates the GPL loses his right of distribution, until such time as that right is restored by affirmative act of the copyright holder. The distributees of that licensor, however, retain their rights under the license, including their rights of distribution. (See Exhibit B, §4).

23. The FSF is by no means the only licensor of programs under the GPL. FSF accepts copyright assignments of some programs for release under GPL, as part of its mission to facilitate the employment and diffusion of free software. The IBM Corporation, for example, not only releases some programs under the GPL, but also assigns copyright to FSF in some of the programs it so releases, for the purpose of empowering FSF to enforce the GPL against license violators. But many other authors of programs choose to release their works under GPL while retaining the ownership of their copyrights: the Linux kernel itself, for example, is owned by its authors; FSF has no significant ownership of copyrights in the primary versions of the Linux kernel. FSF's own enforcement practices in dealing with GPL violations, however, are widely followed in the community of free software and (as it is sometimes called) "open source," a phrase which refers to the requirement that source code be provided to facilitate understanding, modification, and redistribution by all users. FSF's enforcement practices are relevant to the community at large because FSF is the author of the GPL, because it has a large inventory of free software which it manages on behalf of the community as a whole, and because its strong engagement with the ethical as well as commercial value of free software has given it a special position of trust within the community.

24. In my role as General Counsel of the Foundation, I have been primarily responsible for all world-wide enforcement activity in defense of the GPL since 1994. I have been involved in all significant cases in which FSF itself enforced the license with respect to software whose copyrights it held, and I have provided extensive advice to authors of other software released under the GPL with respect to the enforcement of their rights and protection of the integrity of the license. In those activities, I have found §4 of the GPL to be absolutely essential to the conduct of our enforcement strategy.

25. FSF's policy with respect to GPL violations is to secure compliance, not damages. When a party has violated GPL, and the violation is called to our attention (which happens on the average some dozens of times each year), we inform the party in violation of its responsibilities, and advise it on the steps necessary to come into compliance. It is our practice that once a party has taken steps to comply, and has entered into confidence-building measures to ensure that future non-compliance will be avoided wherever possible, and rapidly discovered and remedied where inadvertently reproduced, distribution rights under GPL §4 are restored on a cooperative non-judicial basis. In this fashion, I have secured compliance with the license in dozens of cases over the past decade, and have never had to resort to judicial measures of mandatory enforcement. Without the leverage provided by §4, however, parties would resort to repetitive partial compliance, "capable of repetition but evading review," in language the Supreme Court has applied to a different sort of situation, substantially if not overwhelmingly complicating the task of securing reliable compliance with the license.

### MySQL

26. I understand that MySQL AB was formed by the primary authors and is a copyright holder of the program MySQL, which is a particularly important component among the many thousands of components of the free software system. MySQL is a "database engine," which means that it organizes and manages access to large quantities of "tabular" data.

27. Database tables would include the transactions in a store, the reservations in an airline reservation system, the addresses in a mailing list, or the personnel records of the employees of a company. Each "record"



in such a database can be thought of as the “row” of a table, in which a single sales transaction, for example, contains several “fields,” or “columns” (the item number of the item sold, the quantity, the price, the shipping charge and tax applied, and so on). The “SQL” in the name stands for “structured query language,” which is the historically-conventional phrase used to describe database engines that allow rows in the table to be searched for by a complex set of restrictions on the fields (“Give me all the transactions in which someone in zipcode 10027 bought five or more of item #3116 in color green or blue during the months of April and May 2000,” for example).

28. The “engine” which manages the creation and searching of such tables, particularly when they involve hundreds of thousands or millions of entries, is a critical building block of larger software applications. Most of the “e-commerce” occurring on the World Wide Web, whether it involves making a purchase in a store, reserving transportation or accommodation, or tracking the movement of packages by freight companies, involves interacting with a database manager.

29. MySQL is the most popular and widely-used of free software database engines. Along with the program called Apache that manages websites, and the Perl and PHP “scripting languages” that allow web pages to accept and process user input, MySQL is part of the “platform” on which many thousands of individual applications have been constructed by businesses and other organizations, large and small, around the world.

30. MySQL AB engages in “dual licensing.” This means that it licenses a version of MySQL to be freely used, copied, modified and distributed by everyone under the GPL, and also makes versions of its program that are distributed to particular customers without the right of free distribution. Those who receive MySQL under the GPL, however, are not entitled to engage in “dual licensing.” Having received their copy of the program under GPL, they may freely modify and redistribute, but that redistribution, under GPL §2(b), must occur under the terms of the GPL, without any additional limitation. In particular, anyone who modifies MySQL must release that modified version in compliance with the GPL’s requirement that everyone who receives the program must receive “source code,” or all the materials required so that they themselves can

understand, share and improve the program in its modified form.

31. Progress Software Corp. is the distributor of program code called "Gemini." Gemini is a "storage module" for a database engine. It performs the task of writing data into a new or existing database, which is a subcomponent of the tasks performed by the database engine as a whole. Gemini provides numerous features for any database engine into which it is inserted. Gemini can provide "crash recovery": if a large database stored by Gemini is in the process of being updated when the computer system managing the data is suddenly halted, by a power outage or similar unexpected intervention, Gemini can assure that the data is uncorrupted. Gemini also provides "reversability," in the sense that a change made in the database is not final when it is first made, and thus, if it turns out that the change is erroneous, until the table stored by Gemini is "committed," the original pre-change data can still be recovered. But Gemini tables are also larger than tables stored by other storage managers, and the task of storing information in such tables is slower. For this reason, a database engine such as MySQL relies upon multiple storage modules, rather than only one. When a particular data generator decides how to construct a database for a particular task, it will instruct the database engine to choose among storage modules in order to achieve the optimal balance of size, speed, stability, reversability, and other properties. In any particular computer system that provides database applications, the MySQL engine will be a single program, usually called "mysqld." The mysqld program on that system will have one or more storage modules "compiled" or "linked" into it as subcomponents, like chapters in a single book, and those particular chapters will have been included that serve the needs of the particular application or applications that will use database services.

### **The Current Dispute**

32. In connection with the instant litigation, I have reviewed two versions of "NuSphere MySQL Advantage" distributed by Progress Software Corporation. I have conducted that review precisely as I would have conducted such an investigation in my role as FSF's General Counsel in the event of a complaint of GPL violation. I personally conducted all activities hereinafter described, depending on my own personal knowledge of the practices of software manufacture and distribution, as well as my legal understanding of the

requirements of FSF's GNU General Public License.

33. NuSphere MySQL Advantage is a collection of programs, like a bookshelf containing many books, all of which have been adapted to work smoothly together. Among the separate works anthologized in NuSphere MySQL Advantage are the Apache web server program, the Perl and PHP scripting languages, and MySQL. A user who buys NuSphere MySQL Advantage is thus getting "one-stop" shopping: each of the programs involved is free software, released under a variety of free software licenses, and could each be gotten from multiple places in the network, usually at nominal cost. By paying a substantial price for NuSphere MySQL Advantage, however, the user gets everything all on one CD ROM, arranged to work smoothly together and to be installed using a simple interface that can be controlled from an ordinary web browser, and which will install the same way on machines running Microsoft Windows, GNU/Linux, or other operating systems for non-PC computers.

34. The first version of NuSphere MySQL Advantage I reviewed was labeled Version 2.2. The CD ROM containing the software associated dates with each file in the customary fashion, and by reviewing the dates I concluded that the CD ROM was "burned," or fixed in a tangible form, on June 21, 2001. I installed this copy of NuSphere MySQL Advantage on a computer running GNU/Linux, following the standard directions provided. The standard installation program created a `mysqld` program on my computer's hard drive, and once that program had been created I used a standard information request to inquire of that copy of `mysqld` whether it included Gemini storage capacity. The program told me that Gemini storage capacity was included in `mysqld`, and gave me certain technical particulars, for example the largest Gemini table that it could store.

35. I then checked whether the source code of the Gemini component of `mysqld` was available. I saw that it was not. I used standard tools to "recompile" `mysqld`, which means to rebuild the program from the parts provided, and discovered that the rebuilt `mysqld` did not include Gemini capacity. In a manual distributed in the portion of the CD only related to Windows users I found the statement that "'GEMINI' tables will be included in some future MySQL 3.23.X source distribution."

36. On the basis of this examination, based upon my expertise and prior experience as a computer

programmer, I concluded that the version of NuSphere MySQL Advantage under review violated the requirements of the GNU GPL. The mysqld program as distributed was covered by the GPL. It had been combined with program code from Gemini to provide those storage types, and had been “compiled” into the “machine language” copy of mysqld distributed to buyers. But the source code for the portion of mysqld that performed Gemini storage had been withheld, which violated GPL §3. This omission was not inadvertent, as was shown by the comment in the Windows version of the MySQL documentation, which said only that source code would be provided at a later date. Given my knowledge of the practices of programming and the requirements of the GPL, I concluded that the license violation was intentional. Whether intentional or not, any violation of the GPL results under §4 in a termination of the right to redistribute.

37. I then reviewed a copy of Version 2.3.1 of NuSphere MySQL Advantage. The dates in the filesystem allowed me to conclude that the CD ROM was “burned” on October 1, 2001. Installing as ordinarily directed I again verified that the version of mysqld placed on my hard drive contained Gemini storage capacity. I reviewed the source code directories and found that the source code of the Gemini module was fully available. I then verified that I could “compile” or rebuild mysqld from the available parts so that it matched the version that had been distributed in “machine language” on the disk. The statement about postponed source code availability had been removed from the Windows manual.

38. Under GPL §4, I conclude, Progress Software Corp. lost the right to distribute MySQL when it distributed NuSphere MySQL Advantage in a fashion that violated GPL.

I declare under penalty of perjury and upon personal knowledge that the foregoing is true and correct.

Dated: February 26, 2002  
New York, New York

---

Eben Moglen

# The GNU General Public License (GPL)

## Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

### **TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

**0.** This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included

without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.



7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### **NO WARRANTY**

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND

PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

**12.** IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

#### **How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail. If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to
redistribute it under certain conditions; type `show c' for
details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest  
in the program `Gnomovision' (which makes passes at compilers)  
written by James Hacker.

signature of Ty Coon, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1. Disclosure of Risks of Open Source Software Incorporation, from Form 10-K Filed with U.S. S.E.C. by Traditional, Proprietary Software Vendor

... Factors Affecting our Future Operating Results

...

*Risks Related to Our Intellectual Property*

...

*Our results of operations may decline if we were subject to a protracted infringement claim or one that results in a significant damage award.*

...

In addition, from time to time there have been claims challenging the ownership of open source software against companies that incorporate open source software into their products. We use a limited amount of open source software in our products and may use more open source software in the future. As a result, we could be subject to suits by parties claiming ownership of what we believe to be open source software. Any of this litigation could be costly for us to defend, have a negative effect on our results of operations and financial condition or require us to devote additional research and development resources to change our products. ...

2. Sample Prohibition of Open Source Software "Commingling" In Software/Tech. License Between Traditional For-Profit Proprietary Technology Vendors (extracted from actual, October 2000, disclosed transaction)

... 3.9 ***Use of Identified Software.*** Licensee's rights set forth in this Agreement are conditioned upon Licensee (a) not incorporating Identified Software (as defined below) into or combining Identified Software with the Software Title; (b) not distributing Identified Software in conjunction with the Software Title; and (c) not using Identified Software in the development of the Software Title; in a manner that would affect [LICENSOR]'s rights or obligations with respect to the Software Title. "Identified Software" means software which is licensed pursuant to terms that (i) create, or purport to create, obligations for [LICENSOR] with respect to the Software Title or (ii) grant, or purport to grant, to any third party any rights or immunities under [LICENSOR]'s intellectual property or proprietary rights in the Software Title. Identified Software includes, without limitation, any software that requires as a condition of use, modification and/or distribution of such software that other software incorporated into, derived from or distributed with such software be (a) disclosed or distributed in source code form; (b) be licensed for the purpose of making derivative works; or (c) be redistributable at no charge. ...

3. Sample Prohibitions Of Open Source Software Prior Use or Future Impacts, Extracted From 7/03 or 8/03 Disclosed Merger/Acquisition Agreements Among Technology Corporations

a. One Example.

... (m) Intellectual Property. ...

(viii) None of the software products of the Company or any Subsidiary is subject to the provisions of any open source or other source code license agreement that (A) requires the distribution of source code in connection with the distribution of the licenses software in object code form; (B) prohibits or limits the Company or any Subsidiary from charging a fee or receiving consideration in connection with sublicensing or distributing such licensed software (whether in source code or object code form); or (C) allows a customer or requires that a customer have the right to decompile, disassemble or otherwise reverse engineer the software by its terms and not by operation of law. ...

b. Another Example.

... (j) *Open Source Software*. No Software that is open source, public source or freeware, or any modification or derivative thereof, including any version of Software licensed pursuant to any GNU general public license or lesser general public license (collectively, “**Open Source Software**”) was or is used in, incorporated into, integrated or bundled with any Company Products (only to the Company’s Knowledge with respect to Company Products owned by third parties) or Company Intellectual Property, or was, is or is currently planned to be used in the Company’s business or the development or compilation of any Company Products, except in each case for any such Open Source Software whose terms: (i) do not grant any third party rights to, or otherwise impair the exclusive ownership of the Company or its Subsidiaries of, the modifications or improvements by the Company or any of its Subsidiaries to such Software; (ii) do not require the distribution by the Company or any Subsidiary of any source code; (iii) do not prohibit the Company or its Subsidiaries from charging a fee or otherwise restricts the ability of the Company or any of its Subsidiaries to seek compensation for any such Software; (iv) allow use and distribution of such Software in the manner used and distributed and proposed to be distributed by the Company or any of its Subsidiaries; (v) do not contain any other material restrictions or obligations that the Company and its Subsidiaries has not complied with; and (vi) do not, have not and will not otherwise result in any material liability to the Company or any of its Subsidiaries. ...

**Is Open Okay?  
Managing New Open Source Software  
Resources, Risks and Rules**

American Corporate Counsel Association  
Annual Convention  
“Charting A New Course For You, Your Profession, and Your Corporation”  
Panel 710  
October 9, 2003  
4:30 to 6 p.m.  
San Francisco Marriott

Heather D. Rafter  
General Counsel  
Digidesign

Special thanks to Heather J. Meeker, Tomlinson Zisko, LLP

Note: The views expressed in this document are those of the author and are not meant to be construed as the statements of either Digidesign or Avid Technology, Inc.

## OPEN SOURCE CODE CHECKLIST FOR IN-HOUSE COUNSEL

Overview: This checklist provides an overview of procedures and issues that in-house counsel might want to consider with respect to use of open source code. It is not meant to be exhaustive and, in the event your company is considering use of open source code, independent consultation with counsel, particularly those who are specialists in this area, is highly recommended.

Attached to this checklist is sample attribution and license agreement language to use in connection with licensing of open source code.

1. Establish a policy regarding use of open source code
  - a. Is it permissible? Note: As a practical matter, it may be impractical or unwise to ban all use of open source code; however, the determination should be made on a case-by-case basis in consultation with engineering and the legal department, which should be aware of and have to approve all uses of open source code.
  - b. If an engineer intends to use open source code, who is the person within the legal department to be contacted?
2. Routinely audit to ensure that the in-house legal department is aware of ALL uses of open source code, either as a development tool or in the code base of a product used internally or shipped to customers.
3. Educate the engineering and IS departments regarding open source code and the need to apprise legal of any use. Engineers should keep a log of each open source software module used and print a copy of the license agreement that applies to it at the time of download. This information, in turn, should promptly be disclosed to the legal department.
4. If open source code is being used, need to determine:
  - a. How is the code being used?
    - i. If in development environment, less likely that there are “flow-down” (i.e., relicensing) requirements
    - ii. If used within product, is open source code used directly in code base or linked?

- iii. Some development tools include code libraries that are incorporated into products created with them.
- b. How reliable is your licensor?
- i. Some open source licensors are not in the business of licensing software, so they may not adhere to professional standards in clearing the rights to their code.
  - ii. Beware of authors who have authored code that might be a work for hire (moonlighters, university employees, etc.).
  - iii. Be as careful with clearing rights in open source code as you are with other, proprietary components that you get under inbound licenses.
- c. What are the license requirements?
- i. It's helpful to first classify type of license to understand general rules. License types may include: Apache, BSD, GNU Lesser General Public License (LGPL), GNU GPL 2.0 (also referred to as the "Greater GPL") and other variations.
  - ii. Next step is to review license and understand specific rules governing each license.
    - 1. Are there "flow down" (relicensing) requirements?
    - 2. Is there potential for any viral impact (more likely with the "greater" GNU GPL)?
    - 3. What attribution or other marketing requirements exist?
    - 4. If you are a software company, does use of the open source code require changes to end user license agreements?

**Note:** Each and every license accompanying use of the open source code should be carefully reviewed to ensure full understanding and compliance with its terms. However, the meaning of commonly used open source agreements may be affected by custom and practice in the industry, so consult an expert.



**Sample Attribution Language to insert in marketing materials or the product “About Box”:**

© 2003 [insert your company name]. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Those portions © 2000-2003 The Apache Software Foundation. All rights reserved.

Portions of this product are covered by the GNU Lesser General Public License. Those portions © 1991, 1999 Free Software Foundation, Inc.

This product includes software developed by the Open SSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>). Those portions © 1998-2003 The OpenSSL Project.

The language above combines a copyright notice for your company’s product (first line) with examples of flow-down notice requirements for several popular open source agreements.

**Sample paragraph to add to your End User License Agreement (“EULA”):**

Open Source Portions. Certain portions of the Software (“Open Source Components”) are not licensed under the terms of this Agreement, but are instead licensed under the terms of applicable open source licenses, such as the BSD License, Apache License or the Lesser GNU General Public License. Company grants you no right to receive source code to the Open Source Components; however, in some cases rights and access to source code for the Open Source Components may be available directly from Company’s licensors. **[Upon request, Company will identify the Open Source Components and the licenses that apply to them.] [Information identifying the Open Source Components, and the licenses that apply to them, is available on Company’s Web site at \_\_\_\_\_].** Your use of each Open Source Component is subject to the terms of each applicable license. You must agree to the terms of each such applicable license, or you should not use the Software.

This language represents a pass-through of rights to open source components. Keep in mind that although you may pass through different licensing terms, there may be an expectation that you, as the Company, will stand behind the entire product when you license it to customers or if you sell the rights to an acquiror. Counsel should consider how best to manage this expectation and ways to minimize risk, such as through due diligence (see, for example, Section 4(b) above). You also want to consider the scope of any warranties and indemnifications you may be offering.

# Is Open Okay? Managing New Open Source, Resources, Risks & Rules

## The View from an Open Source Provider

by

**Mark H. Webbink**  
**Sr. Vice President and General Counsel**  
**Red Hat, Inc.**

### Understanding Open Source Software

#### What is Open Source Software?

The Open Source Initiative ("OSI") defines open source as software providing the following rights and obligations:

- 1.No royalty or other fee imposed upon redistribution
- 2.Availability of the source code
- 3.Right to create modifications and derivative works
- 4.May require modified versions to be distributed as the original version plus patches
- 5.No discrimination against persons or groups
- 6.No discrimination against fields of endeavor
- 7.All rights granted must flow through to/with redistributed versions
- 8.The license applies to the program as a whole and each of its components
- 9.The license must not restrict other software, thus permitting the distribution of open source and closed source software together

This definition clearly leaves room for a wide variety of licenses, and we will examine a number of those license types shortly. Although it is this OSI definition of Open Source to which the remainder of this paper relates, it is worthwhile to also examine the definition of Free Software, for often times the terms Free Software and Open Source are used interchangeably. While they are similar, there are differences worth appreciating.

When we speak of Free Software, we are not talking about freeware, i.e., software that is essentially in the public domain. Rather, we are talking about software that is licensed under the precepts of the Free Software Foundation ("FSF") and its flagship GNU General Public License. According to the FSF definition:

“Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

0. The freedom to run the program, for any purpose (freedom 0).
  1. The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
  2. The freedom to redistribute copies so you can help your neighbor (freedom 2).
  3. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. (freedom 3). Access to the source code is a precondition for this.
- A program is free software if users have all of these freedoms.”

Contrasting the Open Source and Free Software definitions, one finds that all Free Software is Open Source, but as administered by the Free Software Foundation, not all Open Source is Free Software. The difference principally arises from so-called license compatibility, but in large measure the differences are principally philosophical and not substantial.

### **Fundamentals of Copyright Law**

To better appreciate Open Source software, we need a basic understanding of copyright law. Open source software is fundamentally grounded in copyright law<sup>1</sup>. In order to appreciate the rights granted under open source licenses, one must first be familiar with the basic bundle of rights granted the holder of a copyright. Under U.S. copyright law, those rights are:

1. The exclusive right to copy the work;
2. The exclusive right to make derivative works;
3. The exclusive right to distribute the work;
4. The exclusive right to perform the work; and
5. The exclusive right to display the work.<sup>2</sup>

These rights, in turn, are subject to certain limitations, such as rights of “fair use.” Fair use includes the use of a work for purposes of criticism, comment, news reporting, teaching, scholarship or research and does not constitute infringement of the work. Whether a specific use is fair use is determined by a number of factors, including: (1) the

---

<sup>1</sup> When I talk about copyright law in this paper, I am discussing U.S. copyright law as embodied in Title 17 of the United States Code. The United States is a signatory to the Berne Convention covering copyright, and much of U.S. copyright law is very similar to that of other Berne signatory countries. However, there are provisions in copyright law in the U.S. that are unique to the U.S., such as copyright registration. Persons in countries other than the U.S. who are relying on information provided in this book should consult local legal counsel specializing in copyright law.

<sup>2</sup> §1-106, Title 17, U.S. Code. Note that when used in the context of copyright law, performance and display refer to public performance and display, not private use.

purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes; (2) the nature of the copyrighted work; (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and (4) the effect of the use upon the potential market for or value of the copyrighted work.<sup>3</sup>

Works, such as software, may be placed in the public domain and exist outside of the scope of copyright law.<sup>4</sup> However, with changes in the copyright law in the 1970's and 1980's, including the automatic application of copyright under the Berne Convention, it is no longer an easy task to contribute software to the public domain.<sup>5</sup> Software (or any other body of work) that is in the public domain cannot, by definition, assert any restrictions on who or how it can be used, modified or distributed (though other laws, such as export controls, may still restrict some software's use or distribution). If open source software were in the public domain (that is, not subject to copyright because the author has disclaimed copyright in the work), any business or individual could use the software for any purpose without any copyright restriction, and there would be no requirements for legal review above and beyond ensuring compliance with other statutes (which apply equally to all other software, public domain, or not). Because Open Source software is not in the public domain, but instead protected by copyright law and licensed for use under certain, perhaps unconventional, terms, those terms must be understood.

A valid copyright license applies to a body of work and must assert at least one restriction. A copyright license that states no restrictions implicitly grants all rights, including rights to use, modify, distribute, etc. Most proprietary software copyright licenses assert restrictions on use (including definitions of "fair use", which, according to such licenses, usually does not include decompiling, reverse engineering, or other such uses), copying (usually only for the purposes of backup), and redistribution (usually only when acting as an authorized agent for the copyright owner).

### **Types of Open Source Licenses**

Open source licenses may be broadly categorized into the following types: (1) those that apply no restrictions on the distribution of derivative works (we will call these Non-Protective Licenses because they do not protect the code from being used in non-open source applications), and (2) those that do apply such restrictions (we will call these Protective Licenses because they ensure that the code will always remain open/free).

---

3 §1-107, Title 17, U.S. Code.

4 37 CFR 201.26 defines public domain computer software as software which has been publicly distributed with an explicit disclaimer of copyright protection by the copyright owner. As the Free Software Foundation has stated, public domain software means software that is not copyrighted.

5 Under the Judicial Improvements Act of 1990, which authorized the creation of a national shareware registry, software copyright owners may donate their software to the public domain by assigning it to the Machine-Readable Collections Reading Room of the Library of Congress. 37 Code of Federal Regulations Part 201.26 (1991)

To better appreciate the nature of these licenses, it is helpful to picture software licenses on a continuum based on the rights in copyright extended to the licensee.

Software that has been placed in the public domain is free of all restrictions, all rights under copyright having been granted to the public at large. Licensors of Non-Protective open source licenses retain their copyright, but they grant all rights under copyright to the licensee. Licensors of Protective open source licenses retain their copyright, grant all rights under copyright to the licensee, but apply at least one restriction, typically that the redistribution of the software, whether modified or unmodified, must be under the same license. Licensors of propriety licenses retain their copyright and only grant a few rights under copyright, typically only the rights to perform and display. The following table, where the BSD license is used as an example of a Non-Protective open source license and the GNU General Public License as an example of a Protective open source license, displays these contrasts:

<b>Rights Granted</b>	<b>Public Domain</b>	<b>BSD</b>	<b>GPL</b>	<b>Windows 98</b>
<b>Copyright retained</b>	<b>No</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
<b>Right to perform</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
<b>Right to display</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
<b>Right to copy</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>
<b>Right to modify</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>
<b>Right to distribute</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes, under same license</b>	<b>No</b>

Non-Protective open source licenses include: Academic Free License v.1.2; Apache Software License v.1.1; Artistic; Attribution Assurance license; BSD License (shown in **Appendix B**); Eiffel Forum License; Intel Open Source License for CDSA/CSSM Implementation; MIT License; Open Group Test Suite License; Q Public License v.1.0; Sleepycat License; Sun Industry Standards Source License; University of Illinois/NCSA Open Source License; Vovida Software License v.1.0; W3C Software Notice and License; X.Net, Inc. License; zlib/libpng License; and Zope Public License v.2.0.

Protective open source licenses include: Apple Public Source License v.1.2; Artistic License; Common Public License v.1.0; GNU General Public License v.2.0; GNU Lesser General Public License v.2.1; IBM Public License v.1.0; Jabber Open Source License v.1.0; MITRE Collaborative Virtual Workspace License ; Motosoto Open Source License v.0.9.1; Mozilla Public License v.1.0 and v.1.1; Nethack General Public License; Noika Open Source

License v.1.0a; OCLC Research Public License v.1.0; Open Software License v.1.1; Python License; Python Software Foundation License v.2.1.1; Ricoh Source Code Public License v.1.0; and Sun Public License v.1.0.

All of these, and additional new licenses, can be found on the Open Source Initiative website ([www.opensource.org](http://www.opensource.org)).

Some open source licenses of both types include other provisions, such as restrictions on the use of trademarks, express grants of license with respect to applicable patents, disclaimers of warranties, indemnification of copyright holders in commercial distributions, and disclaimers of liability. However, none of these provisions are as fundamentally important as the obligations/restrictions that are imposed on redistribution rights under the Protective open source licenses, and it is with those restrictions on redistribution that we next focus.

### **The GNU General Public License**

As of this writing, the GNU General Public License (“GPL”) is the most pervasive license of Open Source software. Of all the software to which it has been applied, none is better known than the Linux<sup>®</sup> kernel. In fact, the GPL has been applied to a majority of those software modules that are included in the best known of the Linux<sup>®</sup> distributions, such as Red Hat<sup>®</sup> Linux<sup>®</sup>. Its wide appeal among the Open Source community stems from the fact that it falls into that category of open source licenses which obligate parties who wish to redistribute such software, either in original or modified (derivative) form, to do so under the terms of the license agreement under which such software was received (all of which we refer to as Protective licenses). That is, having been granted the right to use, modify and redistribute the software under the GPL, the GPL requires you to extend those same privileges under the same terms to others who receive the software from you. This is the common thread that governs Protective licenses, and for that reason, we will focus on the GPL as the standard for Protective licenses.

The GPL, a copy of which may be found in [Appendix A](#), provides certain rights to anyone receiving a license to software governed by the GPL. At the same time, it imposes very few obligations except on those who wish to redistribute the software: Those rights and obligations are:

- The right to copy and redistribute so long as you include a copyright notice and a disclaimer of warranties.  
    You may charge for the cost of distribution and you may offer warranty protection for a fee.
- The right to make derivative works for your own use
- The right to distribute derivative works so long as you
  - Identify the work as modified
  - License it under the GPL
  - Provide the license information interactively if the program normally runs interactively
  - This section, and the obligation to license under the GPL, does not apply to works which are independent works distributed with the GPL'd work and which run on the GPL'd works.

- You may distribute the work only in executable form so long as the source code is
  - distributed with the object code
  - offered by a written offer, valid for a period of at least three years, to make the source code available for no more than the cost of distribution
  - for non-commercial distributions, accompanied with the offer the redistributing party received as to the availability of the source code
- You may not impose restrictions on any of these rights

This is a simple, yet elegant approach. Basically, the licensor is permitting any licensee to exercise virtually all of the rights available under copyright, i.e., the right to copy, the right to make derivative works, the right to distribute, the right to perform, the right to display. The only obligation imposed is, if the licensee, in turn, wishes to distribute the software to other parties, they agree to do so only under the GPL. The sole purpose of these restrictions is to preserve the integrity of the original grant of freedom through any path of redistribution and to make it impossible for anybody to create a version of the software that offers less freedom to any recipient than the original version would have granted. To paraphrase, the GPL states “once free, always free.”

Note that the GPL has no relevance to the case where a party licenses the software and chooses not redistribute it. This is true whether the party is an individual, a corporation, a corporate conglomerate, or the government. As noted by the FSF, when the GPL refers to “You” in the context of a corporation, it means the parent company and all of the controlled subsidiaries of that parent. Similarly, when “You” is addressing a unit of government, it means that unit of government and all of the subdivisions of that government that are under the direct control of that government. In that context, “You” can readily mean the entire federal government of the U.S. or it could mean any state or commonwealth government, including the agencies of that state or commonwealth government.

The GPL does not require that a licensee, who has not made a distribution of the software to another, provide copies of that software to any party who so demands it. The restrictions of the GPL apply only in the case of where GPL'd software is being provided to another party, and the GPL pertains only to the preservation of its original purpose—nothing more.

Based on the foregoing, we can divide the types of open source usage into categories, and analyze the legal implications of the GPL for each category. The interesting categories are:

1. Users who use only GPL binaries as they would any other similar program:
2. Users who modify GPL sources to handle local configuration issues or to address internal requirements and not for distribution to others: and
3. Users who modify GPL sources and redistribute them for fun and/or profit.

In case (1), the GPL affects these users not at all; use of the open source GNU Emacs™ text editor does not imply that the act of saving a file changes the ownership of the file to the FSF, nor does compilation of a file by open source GNU C Compiler cause the resulting object code to belong to the FSF, nor does setting a breakpoint in an executable cause the executable to suddenly become the property of the FSF. Thus, the normal use of GPL software (i.e., use like one would use any other commercial software) in a commercial environment poses no extraordinary legal problems. The wide distribution of Linux operating system software in the last several years for use on commercial web and enterprise servers is ample evidence that there is no legal reason to not use Open Source software if you happen to think it is better than the proprietary alternatives.

In case (2), the locally modified software by definition confers to its users access to the locally modified sources. There is no requirement within the GPL that such local modifications be disclosed to any other party.

In case (3), we get to the group of users for whom the GPL was really written. Users redistributing modified or unmodified versions of Open Source software must obey the GPL's "Golden Rule" of licensing the distributed software under the GPL and not adding any downstream restrictions. To the extent that somebody wants to profit from GPL'd software by using traditional proprietary license restrictions, those restrictions will prove difficult if not impossible to apply. Note, however, earning profit because of the GPL is both legal and encouraged.

From this analysis we are left needing a definitions of what constitutes a derivative work in software.

### **What is a derivative work?**

The U.S. Copyright Act defines a derivative work as:

“a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications, which, as a whole, represent an original work of authorship, is a "derivative work".<sup>6</sup>

Thus, a work that is based on one or more preexisting works constitutes a derivative work to the extent the new material added constitutes an original work of authorship. Such new material may include editorial revisions, annotations, elaborations or other modifications. Derivative works may transform the original work, such as in a translation, including translating software from one computer language to another, or they may combine the original work with other works, such as in a compilation like Red Hat® Linux®. Copyright protection in a derivative work or

---

6 17 U.S. Code §101



compilation extends only to the material contributed by the author of such work, and does not grants rights in preexisting material included in the new work.<sup>7</sup>

### **Where does the law stand on derivative works in software?<sup>8</sup>**

The law on derivative works in software is not well established. The U.S. Copyright Act does not specifically address derivative works in software, and there are no U.S. Supreme Court cases immediately on point. Most of the case law has developed among the various U.S. Courts of Appeals, but even there the law varies from one circuit to the next.

The Copyright Act provides an important definition in addition to that of “derivative works”, that of “computer programs”, which are defined as:

“a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.”<sup>9</sup>

In addition, the Copyright Act limits the scope of what is covered by copyright by excluding certain subject matter. §102(b) of the Act provides:

“In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.”

Perhaps the most established of the tests for derivative works in software is the “abstraction, filtration, and comparison” (“AFC”) test established by the Second Circuit.<sup>10</sup> Under the three-part AFC test, a court first determines (abstracts) the constituent structural parts of the original program. From these structural parts, the court then filters all unprotectable portions, including those unprotectable matters defined in §102(b) of the Copyright Act and elements that are in the public domain. In the final step, the Court compares any remaining code containing creative expression to the structure of the second program to determine whether the software program in question is sufficiently similar to the pre-existing work as to justify a finding that the second program is a derivative work of the first. This AFC approach has been adopted by three other circuits: the Fifth<sup>11</sup>; Tenth<sup>12</sup> and Eleventh.<sup>13</sup>

---

7 17 U.S. Code §103

8 For an in depth discussion of the state of the law, see “Software Derivative Work: A Circuit Dependent Determination”, Dan Ravicher, October 31, 2002, [http://www.pbwt.com/Attorney/files/ravicher\\_1.pdf](http://www.pbwt.com/Attorney/files/ravicher_1.pdf).

9 17 U.S. Code §101.

10 *Computer Associates Intl. V. Altai, Inc.*, 982 F.2d 693 (2<sup>nd</sup> Cir. 1992).

11 *Engineering Dynamics, Inc. v. Structural Software, inc.*, 26 F.3d 1335 (5<sup>th</sup> Cir. 1994); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527 (5<sup>th</sup> Cir. 1994).

Of the remaining nine U.S. Courts of Appeal, only one has adopted a clear test for derivative works in software. The Ninth Circuit's test is based on analytical dissection, which first considers whether there are substantial similarities in both the ideas and expressions of the two works and then utilizes analytic dissection to determine whether any similar features are protected by copyright.<sup>14</sup> The similar elements are categorized by the degree of protection they are to be afforded. "Thin" protection is afforded to non-copyrightable facts or ideas that derive copyright protection only from the manner in which those facts or ideas are aligned and presented. "Broad" protection is afforded to copyrightable expression. The court uses these standards to make a subjective comparison of the works to determine whether, as a whole, they are sufficiently similar to justify a finding that one is a derivative work of the other.

### **How do these tests apply to derivative works in open source software?**

In addressing derivative works, Open Source software requires special consideration. This is due principally from the fact that Open Source software, by definition, permits the making of derivative works. Under a Non-Protective license, the new portions of such a derivative work may be licensed under the license of choice of the author, and there is little likelihood of an infringement dispute.

The case is much different with Protective licenses because it requires derivative works to be licensed under the same license as the original work. Here the question largely becomes one degree of copying versus adequate avoidance of derivation. Where Open Source software licensed under a Protective license appears to have been copied, in whole or in part, into a larger work, which is then licensed under a different license than the original work, the question of derivative work and infringement would be determined by the courts using the tests outlined above. However, this is not the case where the subsequent author maintains the original Protective license with respect to the original work but licenses the new work under a different license, for here the subsequent author has not infringed the rights of the original author except to the extent the new work can be determined to be a derivative work of the original. This latter instance requires an entirely different approach to determining derivation.

Where the original work continues to be licensed under a Protective license and the new work is licensed under an alternative license, the following factors are to be considered when determining whether the new work is a derivative of the original:

1. The substantiality of the new work;

---

<sup>12</sup> *Gates Rubber Co. v. Bando Chem. Indust. Ltd.*, 9 F.3d 823 (10<sup>th</sup> Cir. 1993); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366 (10<sup>th</sup> Cir. 1997).

<sup>13</sup> *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11<sup>th</sup> Cir. 1996); *Mitek Holdings, Inc. v. Arce Engineering Co., Inc.*, 89 F.3d 1548 (11<sup>th</sup> Cir. 1996).

<sup>14</sup> *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9<sup>th</sup> Cir. 1994).

2. Whether any part of the original work has been modified; and
3. How such modification has been accomplished.

This analysis is consistent with the distinction drawn by the GPL itself. Clause 2 of the GPL states:

“Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of storage or distribution medium does not bring the other work under the scope of this license.”

For example, if the work in question is a database written entirely by you, and the Program in question is a GPL'd operating system (one of many to which the database may have been ported), the distribution of the database with the operating system on a volume of storage (such as the system hard disk) would not confer the GPL of the operating system to the database software. On the other hand, if modifications are made to the Program (the Operating System) in order to accommodate the Work (the database), then those modifications, which are a derivative work of the Program, would need to be made available under the GPL. No modifications to the Work (the database) need be redistributed in this case.

In summary, the legal requirements of the GPL are quite straightforward for commercial software providers: if you want to use a proprietary revenue capture model, keep your works (i.e., the code) separate from GPL'd works, keep the modifications made to each fully independent, and there will be no problems protecting your primary works. At the same time, any modifications you make to software that is already covered by the GPL will be subject to the GPL.

### **Myths About Open Source**

Before leaving this discussion of Open Source licensing it is worthwhile to address some of the myths or misconceptions that have arisen around Open Source, particularly those that are promoted by one particular proprietary software company.

#### ***Myth 1 - Open Source Software is “viral” and undermines intellectual property rights.***

This myth is particularly rich. First, as already noted, Open Source Software is fundamentally grounded in copyright law. As with the holder of any copyright, the copyright holder for a piece of Open Source Software gets to elect which rights he/she will grant to others. Open Source authors simply choose to grant more rights than proprietary vendors. The mere fact that an Open Source author using a Protective license insists that derivative works that are distributed to others be licensed under the same license should be contrasted with proprietary

software licenses that simply deny the licensee the right to create derivative works or to redistribute them. Each is an exercise in intellectual property rights, and neither is wrong.

***Myth 2 - Open Source Software is more prone to claims of intellectual property infringement.***

The suggestion of the proprietary vendor is that, because the Open Source development model relies on a vast network of Open Source developers who are not necessarily under the control of the distributor, the code produced is far more likely to be exposed to intellectual property infringement claims. The facts simply do not bear this out. While there undeniably have been such claims against some Open Source development projects and/or distributors, the claims have been few and far between. Contrast this with Microsoft, the largest of proprietary software vendors, which has been sued for patent infringement more than a dozen times in the past five years alone. Closed source does not constitute protection against infringement claims.

***Myth 3 - Unlike proprietary vendors, Open Source Software vendors do not provide warranties or indemnity against intellectual property infringement.***

That is true, but no more true for Open Source vendors than for proprietary vendors. For comparison, examine the Windows 98 license in [Appendix C](#) and note that it expressly disclaims any warranty of non-infringement. Trustworthy computing indeed!

***Myth 4 - The GNU General Public License is risky because it has never been tested in court.***

True again. But which is riskier, licensing practices that are constantly being challenged or those that, in their simplicity and effectiveness, have avoided challenge.

***Myth 5 - Making your source code viewable to some users is the equivalent of Open Source.***

Open Source provides value to its customers and users by giving them total control over their computing environments. The customer gets to choose whether to run the standard version or whether modifications are desirable. The customer can not only see the bugs, he/she can fix the bugs. Making source code merely viewable to a few users does not help them understand the code, does not let them modify the code, and most importantly, does not let them fix the code when it breaks. This approach to source code "sharing" equates to entering a public library only to find there is no card catalog and all of the books are in locked glass cases. Yes, you can root around a find the titles of the books, but you have no ability to gain knowledge from them. Proprietary software seeks to maximize its value solely in monetary terms by achieving a monopoly. Open Source software maximizes its value by assuring that a monopoly cannot be achieved.

***Myth 6 - Open Source methods do not produce innovation.***

This myth is promoted by the very company that did not invent the DOS operating system, that did not invent the word processor, that did not invent spreadsheet software, and that did not invent presentation graphics. It is perpetuated against an Open Source community that: (a) developed the Apache webserver which is used to run the majority of web servers in the world today; (b) developed Sendmail, the most popular e-mail management software; and (c) developed BIND, the basis for using domain names instead of IP addresses to locate websites. Clearly, Open Source is capable of advancing the art of software.

Without belaboring this point, let us turn to best practices that a corporate law office should maintain with respect to software, whether Open Source or proprietary.

**Corporate Law Office Best Practices for Software**

As with any form of intellectual property, there are risks associated with the licensing a use of software. Some of those risks may relate specifically to Open Source software, but most often they relate to all software, regardless of the form of license. Following are a series of best practices that every corporate law office should implement across their company:

- 1. Do not permit the uncontrolled importation of software onto company computers.** Do not permit employees to download freeware, shareware, or Open Source software onto company computers without first clearing the license terms with the legal department. At the same time, bar the use of proprietary software except to the extent that the company can account for the permitted licenses. In other words, know what you are putting on your machines--to do otherwise exposes your company to risk.
- 2. Deal with reputable software vendors with financial staying power.** One of the biggest risks a company takes is adopting software that has no future. Equally true is licensing software from a company without the financial wherewithal to maintain and protect that software. Know your vendors. Know their financial strength, know their policies on licensing, know their responsiveness, and know that their software is reliable.
- 3. Know how the software will be used.** It's one thing if Open Source is to be used as an operating system on a backoffice server, it is something altogether different if that same Open Source software is to be modified and embedded in a product. The former is not problematic; the latter may be. At the same time, make sure your IT folks are well aware of the typical proprietary restrictions which prohibit reverse engineering or modification. While some proprietary vendors may permit such activities under a special development license or a community source code license, they do not generally permit the activities under their general commercial licenses. It may be worthwhile to categorize each item of software and its permitted uses, e.g., approved for general use in executable form only, approved for use at the source code level in specialized applications or modified applications, and not approved for any

use. Finally, nature of use is important in knowing whether the software will be distributed outside the company, potentially triggering Open Source licensing restrictions.

**4. Have a means for documenting what software, and what version of that software, is in use.** Knowing this information and having ready access to it will help assure licensing compliance and at the same time permit IT managers the ability to manage the IT architecture and its advancement.

**5. Require documentation of all internal software development projects.** This includes modification of Open Source software. Such documentation should indicate the source of any base software that is modified, all of the authors of the developed software, prior projects (both internal and with prior employers) on which such developers worked, and the identification of any known related intellectual property, particularly patents.

These are but a few suggestions. They are meant to address those issues most commonly found in software, including Open Source software.

For those interested in learning more about Open Source, the following websites are suggested reading:

Free Software Foundation

<http://www.fsf.org>

Open Source Initiative

<http://www.opensource.org>

Technical FAQs on Linux from IBM

<http://www-106.ibm.com/developerworks/linux/library/l-faq/?open&l=252.t=grl.p=LinuxFAQ>

Link to whitepapers on the legality of the GPL

<http://www.newtechusa.com/Viewpoints/GPLLegalityLinks.asp>

Quick Reference for Choosing a Free Software License

[http://zooko.com/license\\_quick\\_ref.html](http://zooko.com/license_quick_ref.html)

Why Open Source Software

[http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)

Linux Security

<http://www.linuxsecurity.com>

## APPENDIX A

### **The GNU General Public License (GPL)**

#### **Version 2, June 1991**

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)



These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### **NO WARRANTY**

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### **END OF TERMS AND CONDITIONS**

##### **How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

**APPENDIX B****BSD LICENSE****The BSD License (As modified removing advertising clause)**

<OWNER> = Regents of the University of California

<ORGANIZATION> = University of California, Berkeley

<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## APPENDIX C

### Proprietary – Microsoft Windows 98 License

#### END-USER LICENSE AGREEMENT FOR MICROSOFT WINDOWS 98

IMPORTANT: READ CAREFULLY: This Microsoft End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the Microsoft software product identified above, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). The SOFTWARE PRODUCT also includes any updates and supplements to the original SOFTWARE PRODUCT provided to you by Microsoft. Any software provided along with the SOFTWARE PRODUCT that is associated with a separate end-user license agreement is licensed to you under the terms of that license agreement. By installing, copying, downloading, accessing or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT; you may, however, return it to your place of purchase for a full refund.

#### SOFTWARE PRODUCT LICENSE

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

#### 1. GRANT OF LICENSE.

This EULA grants you the following rights:

##### \* Systems Software.

You may install and use one copy of the SOFTWARE PRODUCT on a single computer, including a workstation, terminal or other digital electronic device ("COMPUTER").

##### \* Storage/Network Use.

You may also store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT on your other COMPUTERS over an internal network; however, you must acquire and dedicate a license for each separate COMPUTER on or from which the SOFTWARE PRODUCT is installed, used, accessed, displayed or run. A license for the SOFTWARE PRODUCT may not be shared or used concurrently on different COMPUTERS. Notwithstanding the foregoing, any number of COMPUTERS may access or otherwise utilize the file and

print services and peer web services of the SOFTWARE PRODUCT. In addition, you may use the "Multiple Display" feature of the SOFTWARE PRODUCT to expand your desktop as described in the on-line Help file, without obtaining a license for each display.

\* License Pak.

If this package is a Microsoft License Pak, you may install and use additional copies of the computer software portion of the SOFTWARE PRODUCT up to the number of copies specified above as "Licensed Copies".

## 2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

\* Academic Edition Software.

If the SOFTWARE PRODUCT is identified as "Academic Edition" or "AE," you must be a "Qualified Educational User" to use the SOFTWARE PRODUCT. If you are not a Qualified Education User, you have no rights under this EULA. To determine if you are a Qualified Educational User, please contact the Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399 or the Microsoft subsidiary serving your country.

\* Not For Resale Software.

If the SOFTWARE PRODUCT is labeled "Not For Resale" or "NFR," then, notwithstanding other sections of this EULA, your use of the SOFTWARE PRODUCT is limited to use for demonstration, test, or evaluation purposes and you may not resell, or otherwise transfer for value, the SOFTWARE PRODUCT.

\* Limitations on Reverse Engineering, Decompilation, and Disassembly.

You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

\* Separation of Components.

The SOFTWARE PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer.

\* Rental.

You may not rent, lease or lend the SOFTWARE PRODUCT.

\* Trademarks.

This EULA does not grant you any rights in connection with any trademarks or service marks of Microsoft.

\* Application Sharing.

The SOFTWARE PRODUCT may contain Microsoft NetMeeting, a product that enables applications to be shared between two or more computers, even if an application is installed on only one of the computers. You may use this technology with all Microsoft application products for multi-party conferences. For non-Microsoft applications, you should consult the accompanying license agreement or contact the licensor to determine whether application sharing is permitted by the licensor.

\* Support Services.

Microsoft may provide you with support services related to the SOFTWARE PRODUCT ("Support Services"). Use of Support Services is governed by the Microsoft policies and programs described in the user manual, in "on line" documentation and/or other Microsoft-provided materials. Any supplemental software code provided to you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this EULA. With respect to technical information you provide to Microsoft as part of the Support Services, Microsoft may use such information for its business purposes, including for product support and development. Microsoft will not utilize such technical information in a form that personally identifies you.

\* Software Transfer.

The initial user of the SOFTWARE PRODUCT may make a one-time permanent transfer of this EULA and SOFTWARE PRODUCT only directly to an end user. This transfer must include all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades, this EULA, and, if applicable, the Certificate of Authenticity). Such transfer may not be by way of consignment or any other indirect transfer. The transferee of such one-time transfer must agree to comply with the terms of this EULA, including the obligation not to further transfer this EULA and SOFTWARE PRODUCT.

\* Termination.

Without prejudice to any other rights, Microsoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

### 3. UPGRADES.

If the SOFTWARE PRODUCT is labeled as an upgrade, you must be properly licensed to use a product identified by Microsoft as being eligible for the upgrade in order to use the SOFTWARE PRODUCT. A SOFTWARE PRODUCT labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this EULA. If the SOFTWARE PRODUCT is an upgrade of a component of a package of



software programs that you licensed as a single product, the SOFTWARE PRODUCT may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

#### 4. COPYRIGHT.

All title and intellectual property rights in and to the SOFTWARE PRODUCT (including but not limited to any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the SOFTWARE PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PRODUCT are owned by Microsoft or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by Microsoft.

#### 5. DUAL-MEDIA SOFTWARE.

You may receive the SOFTWARE PRODUCT in more than one medium. Regardless of the type or size of medium you receive, you may use only one medium that is appropriate for your single computer. You may not use or install the other medium on another computer. You may not loan, rent, lease, lend or otherwise transfer the other medium to another user, except as part of the permanent transfer (as provided above) of the SOFTWARE PRODUCT.

#### 6. BACKUP COPY.

After installation of one copy of the SOFTWARE PRODUCT pursuant to this EULA, you may keep the original media on which the SOFTWARE PRODUCT was provided by Microsoft solely for backup or archival purposes. If the original media is required to use the SOFTWARE PRODUCT on the COMPUTER, you may make one copy of the SOFTWARE PRODUCT solely for backup or archival purposes. Except as expressly provided in this EULA, you may not otherwise make copies of the SOFTWARE PRODUCT or the printed materials accompanying the SOFTWARE PRODUCT.

#### 7. U.S. GOVERNMENT RESTRICTED RIGHTS.

The SOFTWARE PRODUCT and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software--Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Microsoft Corporation/One Microsoft Way/Redmond, WA 98052-6399.

#### 8. EXPORT RESTRICTIONS.

You agree that you will not export or re-export the SOFTWARE PRODUCT to any country, person, entity or end user subject to U.S. export restrictions. You specifically agree not to export or re-export the SOFTWARE PRODUCT: (i) to any country to which the U.S. has embargoed or restricted the export of goods or services, which currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria, or to any national of any such country, wherever located, who intends to transmit or transport the products back to such country; (ii) to any end-user who you know or have reason to know will utilize the SOFTWARE PRODUCT or portion thereof in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any end-user who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government.

#### 9. NOTE ON JAVA SUPPORT.

THE SOFTWARE PRODUCT MAY CONTAIN SUPPORT FOR PROGRAMS WRITTEN IN JAVA. JAVA TECHNOLOGY IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED, OR INTENDED FOR USE OR RESALE AS ON-LINE CONTROL EQUIPMENT IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF JAVA TECHNOLOGY COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

#### MISCELLANEOUS

If you acquired this product in the United States, this EULA is governed by the laws of the State of Washington.

If you acquired this product in Canada, this EULA is governed by the laws of the Province of Ontario, Canada. Each of the parties hereto irrevocably attorns to the jurisdiction of the courts of the Province of Ontario and further agrees to commence any litigation which may arise hereunder in the courts located in the Judicial District of York, Province of Ontario.

If this product was acquired outside the United States, then local law may apply.

Should you have any questions concerning this EULA, or if you desire to contact Microsoft for any reason, please contact Microsoft, or write:

Microsoft Sales Information Center  
One Microsoft Way

Redmond, WA 98052-6399.

#### LIMITED WARRANTY

##### LIMITED WARRANTY.

Microsoft warrants that (a) the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and (b) any Support Services provided by Microsoft shall be substantially as described in applicable written materials provided to you by Microsoft, and Microsoft support engineers will make commercially reasonable efforts to solve any problem. To the extent allowed by applicable law, implied warranties on the SOFTWARE PRODUCT, if any, are limited to ninety (90) days. Some states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

##### CUSTOMER REMEDIES.

Microsoft's and its suppliers' entire liability and your exclusive remedy shall be, at Microsoft's option, either (a) return of the price paid, if any, or (b) repair or replacement of the SOFTWARE PRODUCT that does not meet Microsoft's Limited Warranty and that is returned to Microsoft with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE PRODUCT has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE PRODUCT will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. Outside the United States, neither these remedies nor any product support services offered by Microsoft are available without proof of purchase from an authorized international source.

##### NO OTHER WARRANTIES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, MICROSOFT AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE PRODUCT, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

##### LIMITATION OF LIABILITY.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF

BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT OR THE FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, MICROSOFT'S ENTIRE LIABILITY UNDER ANY PROVISION OF THIS EULA SHALL BE LIMITED TO THE GREATER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE PRODUCT OR U.S.\$5.00; PROVIDED, HOWEVER, IF YOU HAVE ENTERED INTO A MICROSOFT SUPPORT SERVICES AGREEMENT, MICROSOFT'S ENTIRE LIABILITY REGARDING SUPPORT SERVICES SHALL BE GOVERNED BY THE TERMS OF THAT AGREEMENT. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.