



CHIEF INFORMATION OFFICER

DEPARTMENT OF DEFENSE

6000 DEFENSE PENTAGON
WASHINGTON, D.C. 20301-6000

JAN 24 2022

MEMORANDUM FOR SENIOR PENTAGON LEADERSHIP
COMMANDANT OF THE COAST GUARD
COMMANDERS OF THE COMBATANT COMMANDS
DEFENSE AGENCY AND DOD FIELD ACTIVITY DIRECTORS

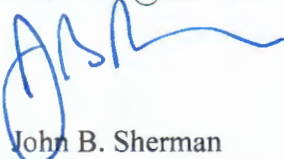
SUBJECT: Software Development and Open Source Software

Over the last two decades, open source software (OSS) has dramatically impacted how software is designed, developed, deployed, and operated. OSS is software for which the human-readable source code is available for use, study, re-use, modification, enhancement, and re-distribution by the users of such software. There are millions of publicly-available OSS components, libraries, and applications capable of accelerating software modernization activities.

The Department's 2018 Cyber Strategy (attached) directed the Department to increase the use of secure OSS and to use commercial off-the-shelf tools when possible. The Department's forthcoming Software Modernization Strategy centers on the delivery of resilient software capability at the speed of relevance. OSS forms the bedrock of the software-defined world and is critical in delivering software faster. The Department must clearly articulate how, where, and when it participates, contributes, and interacts with the broader OSS community.

There are two fundamental concerns for the Department that are specific to OSS. First, using externally maintained code in critical systems potentially creates a path for adversaries to introduce malicious code into DoD systems. This concern requires a careful supply chain risk management (SCRM) approach for OSS, which must meet the same rigorous standards for SCRM and cyber threat testing as any other product. Second, imprudent sharing of code developed for DoD systems potentially benefits adversaries by disclosing key innovations. This risk is managed through a Modular, Open-Systems Approach (MOSA), which allows systems to benefit from OSS while protecting critical, innovative components as separate modules.

Pursuant to Federal Source Code Policy (reference (b)) and Public Law 115-91, Section 875 (reference (c)), Attachment 2 provides detailed guidance on the Department's participation, contribution, and interaction with the broader OSS community. Additional guidance concerning OSS is available at <https://dodcio.defense.gov/Open-Source-Software-FAQ/>. The point of contact for this effort is Dan Risacher, daniel.r.risacher.civ@mail.mil.



John B. Sherman

**CLEARED
For Open Publication**

Jan 26, 2022

Attachments:
As stated

Department of Defense
OFFICE OF PREPUBLICATION AND SECURITY REVIEW

ATTACHMENT 1
REFERENCES

- (a) Department of Defense Cyber Strategy, July 13, 2018
- (b) Office of Management and Budget M-16-21, “Federal Source Code Policy: Achieving Efficiency, Transparency, and Innovation through Reusable and Open Source Software,” August 8, 2016
- (c) National Defense Authorization Act for Fiscal Year 2018, Public Law 115-91, Section 875, “Pilot Program for Open Source Software,” December 12, 2017
- (d) DoD Chief Information Officer Memorandum, “Clarifying Guidance Regarding Open Source Software (OSS),” October 16, 2009 (hereby superseded)
- (e) United States Code, Title 10, Section 2377, “Preference for commercial products and commercial services”
- (f) Federal Acquisition Regulation (FAR), Sections 2.101, 12.000, 12.101
- (g) Defense FAR Supplement, Section 227.7202, “Commercial computer software and commercial computer software documentation” and Section 252.227-7014, “Rights in Noncommercial Computer Software and Noncommercial Computer Software Documentation”
- (h) Federal Acquisition Regulation, Section 13.104, “Promoting competition”
- (i) United States Code, Title 41, Section 3306, “Planning and solicitation requirements”
- (j) Federal Acquisition Regulation, Section 10.001, “Policy”
- (k) National Institute of Standards and Technology White Paper, “Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF),” April 23, 2020
- (l) National Defense Authorization Act for Fiscal Year 2019, Public Law 115-232, Section 1655, “Mitigation of risks to national security posed by providers of information technology products and services who have obligations to foreign governments”
- (m) DoD Instruction 5000.87, “Operation of the Software Acquisition Pathway,” October 2, 2020
- (n) United States Code, Title 10, Section 2322a, “Requirement for consideration of certain matters during acquisition of noncommercial computer software”
- (o) DoD Instruction 5230.24, “Distribution Statements on Technical Documents,” October 15, 2018
- (p) United States Code, Title 10, Section 2446a, “Requirement for modular open systems approach in major defense acquisition programs; definitions”
- (q) Executive Order 13526, “Classified National Security Information,” December 29, 2009
- (r) DoD Instruction 5200.48, “Controlled Unclassified Information,” March 6, 2020
- (s) DoD Cloud Computing Security Requirements Guide, Version 1, Release 3, March 6, 2017
- (t) DoD Instruction 8531.01, “Vulnerability Management,” September 15, 2020

ATTACHMENT 2
GUIDANCE ON SOFTWARE DEVELOPMENT
AND OPEN SOURCE SOFTWARE

1. GENERAL. This attachment provides guidance on OSS and the implications for DoD software development. Generally, custom software is constructed from pre-existing components. Since there are millions of off-the-shelf OSS components available, how the Department uses OSS has a significant impact on overall DoD software development.
2. USE OF OPEN SOURCE SOFTWARE
 - A. The Department must follow an “Adopt, Buy, Create” approach to software, preferentially adopting existing government or OSS solutions before buying proprietary offerings, and only creating new non-commercial software when no off-the-shelf solutions are adequate.
 - (1) OSS meets the definition of “commercial computer software” and therefore, shall be given equal consideration with proprietary commercial offerings, in accordance with Section 2377 of Title 10, U.S.C. (reference (e)) (see also FAR 2.101(b), 12.000, 12.101 (reference (f)); and DFARS 212.212, DFARS 208.74, DFARS 227.7202, and 252.227-7014(a)(1) (reference (g))).
 - (2) In accordance with FAR 13.104, (reference (h)) refusal to consider all OSS based solely on software being open source may be contrary to statutory and regulatory preferences for commercial products, and would unnecessarily restrict competition. OSS should be considered to the maximum extent practical.
 - B. Program managers are ultimately responsible for the suitability of off-the-shelf components used in their programs. This responsibility includes managing risks that the use of these components may introduce to an acceptable level. To the extent that the selection of components and assessment of suitability is delegated to a system integrator, program managers should establish accountability for these functions through contractual language, MOA / MOU, or other directive guidance for government integrators.
 - (1) Agencies are required to conduct market research when assessing and selecting software components per Section 3306 of Title 41, U.S.C. (reference (i)) and Federal Acquisition Regulation 10.001 (reference (j)). When conducting research and determining suitability, factors specific to OSS that require consideration include the following:
 - a. The continuous and broad peer-review enabled by publicly available source code supports software reliability and security efforts through the identification and elimination of defects that might otherwise go unrecognized by a more limited core development team.
 - b. The unrestricted ability to modify software source code enables the Department to respond more rapidly to changing situations, missions, and future threats.
 - c. Reliance on a particular software developer or vendor (“vendor lock-in”) due to proprietary restrictions may be reduced by the use of OSS, which can be operated and maintained by multiple vendors, thus making it easier to replace and upgrade

components as technology and mission needs change. At some level, lock-in may be likely, based on product, architecture, or platform constraints, in spite of using OSS.

- d. Since OSS typically does not have a per-seat licensing cost, it can provide a cost advantage in situations where many copies of the software may be required and can mitigate risk of cost growth in licensing for situations where the total number of users may not be known in advance.
 - e. By sharing the responsibility for maintenance of OSS with other users, the Department can benefit by reducing the total cost of ownership for software, particularly compared with software for which the Department has sole responsibility for maintenance (e.g., Government Off the Shelf (GOTS)). In order to achieve this benefit, the Department must avoid creating a unique version of the software and should participate with the OSS community that supports that software as a user and/or as a contributor.
 - f. OSS is particularly suitable for rapid prototyping and experimentation, where the ability to “test drive” the software with minimal costs and administrative delays can be important.
- C. Because of the collaborative nature of OSS development, supply chain risk management has unique challenges for OSS components. Program managers, in consultation with support engineers, Authorizing Officials, and agency Chief Information Officers (CIOs) must consider these challenges when determining suitability of OSS:
- (1) Long-Term Support: Because of the wide array of business models for creating and maintaining OSS compared to proprietary software, a suitability analysis for OSS must consider risk factors that indicate whether a software module will be adequately supported over the life of the program and how those risks could be mitigated. An analysis of risk factors is essentially an assessment of the "health" of the open source project that maintains that component. For example, “is the OSS project active and/or stable,” “when was the last time this OSS project was updated,” and “who is contributing to this OSS.” Risk mitigations might include consideration of commercial or in-house support for the OSS component, or availability of feasible alternatives (MOA) if the component should need replacement in the future.
 - (2) Trusted Sources: The collaborative development model of OSS often results in many versions of the same software being available from disparate sources. Before using OSS, careful consideration must be made regarding the source from which that software is obtained, and a determination of the trustworthiness of that source. For example, software that is actively maintained and distributed by an established consortium or commercial entity is generally lower risk than a version of that component distributed by an untrusted individual. Adopting a never trust, always verify using a rigorous software assurance approach reduces risk as well. For even trusted sources, program managers should maintain continuous awareness of source compromises and be prepared to respond to sudden loss of trust in a repository. Testing should include dynamic code analysis and penetration testing after integration of the updated versions to ensure that updates do not expose DoD systems to new vulnerabilities.

- (3) **Dependencies:** Because of its modular nature, software is often dependent on sub-components, (e.g., libraries, plug-ins, extensions, and other package modules) that may introduce additional risks. This is particularly true for OSS, which often reuses existing components. Software is higher risk, for example, if it relies on versions of sub-components that are out of date or have publicly known vulnerabilities. Assessment of a software component should also consider security risks of the sub-component dependencies, as well as any legal issues regarding the licenses of sub-components. A related concern is the availability of information related to component libraries, dependencies, and sub-components (such as a software bill-of-materials) that can be used to evaluate such risks.
 - (4) **Component Security:** Any software project that actively works to reduce security vulnerabilities is less risky. Because of the open development process used to develop OSS, it can be easier to assess evidence of this activity (or the lack of it) compared to closed development processes. Positive indicators include the use of tools, both out-of-band and in the OSS project's integration pipeline that look for security vulnerabilities (to detect and fix issues before deployment), transparent reporting of security vulnerabilities, history of security reviews, cyber testing (especially third-party audits, tests, or assessments), problems-addressed, and a history of timely vulnerability remediation. The NIST SSDF (reference (k)) includes additional guidance on component security.
 - (5) **Component Integrity:** Any software project that actively maintains cryptographically protected integrity verification for released code, scripts, configuration files, and associated documentation to reduce security risks (e.g., digitally signed hashes of code) is less risky. Software integrity is particularly critical for OSS programs, which because of the availability of source code, are more subject to the creation and distribution of modified, possibly-untrusted versions.
 - (6) **Influence of Foreign Governments:** Public Law 115-232, Section 1655 (reference (l)) requires certain actions to mitigate risks posed by suppliers of information technology and services who have obligations to foreign governments. OSS is specifically exempt from these requirements, but program managers must be cognizant of the potential for influence on OSS projects by foreign governments, and consider what internal reviews of code contributions are conducted by the OSS project, and what external reviews or audits may be necessary to counteract the potential for malicious interference with the project.
- D. Despite widespread misperceptions that OSS is “free to use”, most OSS is protected by copyright, and the Government's right to use copyrighted software is limited by the terms of the software license. Government use or distribution of OSS must conform to the terms of the OSS license, including maintaining authorship, copyright markings, and licensing information as specified in the license terms.
- E. Given the value of OSS as a source of potential components to software development activities, operators of network enclaves that support software development and maintenance activities should ensure that access to software repositories and software development resources are not unduly restricted. This access is particularly critical for software developers, testers, and analysts to be effective.

F. OSS components of DoD systems require product support and sustainment planning as much as proprietary components. OSS components should be included in the Life Cycle Product Support Strategy (PSS), as described in DoDI 5000.87 (reference (m)), or in a comparable PSS used in other Adaptive Acquisition Framework pathways. For OSS, this strategy should address the ongoing balance between development contractor support, organic contractor logistics support, and support of the OSS community. Many OSS components are updated much more frequently than other COTS or GOTS software, which may impact update schedules for government systems in which they are used. The PSS should consider supportability risks for OSS components over the expected life cycle of the program.

3. USE OF OPEN SOURCE DEVELOPMENT APPROACH

A. In order to release custom-developed (i.e., non-commercial) software as OSS, the DoD must first take delivery of the source code for that software. The Federal Source Code Policy (reference (b)) and Public Law 115-91, Section 875 (reference (c)) require the DoD to release some custom-developed software as OSS. Likewise, in accordance with Section 2322a of Title 10, U.S.C. (reference (n)), programs acquiring non-commercial computer software should plan to acquire and take delivery of the software source code and related materials to the maximum extent practicable.

(1) Program managers should plan to acquire source code of non-commercial software, related data, and associated license rights that are necessary across the life-cycle of that software for the development, testing, operations, and maintenance of that software. This includes all artifacts necessary to reproduce, build, or recompile the software from its source code and required software libraries; to conduct required functional and vulnerability software testing; and to deploy computer programs on relevant system hardware.

(2) Contracting officers are responsible for ensuring that, wherever practicable, solicitations and contracts require delivery in a digital format compatible with applicable computer programs on relevant system hardware and do not rely on external or additional non-commercial or commercial computer software or data that is not also included in the items to be delivered.

(3) Updates to the Defense Federal Acquisition Regulation Supplement with specific guidance are pending as DFARS Case 2018-D018.

B. Managers of government software development projects should separate the development of software that implements critical technology from routine automation of business processes or similar functionality. Critical technology is defined in DoD Directive 5230.24 (reference (o)) as information and technical data that advance current technology or describe new technology in an area of significant or potentially significant military application or that relate to a specific military deficiency of a potential adversary.

(1) The separation of software implementations of critical technology from routine automation should be reflected in a modular open systems approach as defined by Section 2446a of Title 10, U.S.C. (reference (p)). Entire programs generally should not be protected as critical technology, but rather only the components that are actually novel advances.

- (2) Software development that implements critical technology (as defined by reference (o)) is generally not appropriate for distribution as OSS. If shared at all, it must be closely protected and shared only with mission partners under a non-disclosure agreement with terms to protect its strategic value. OSS may be used as a component of a critical technology unless the license terms of the OSS component will create an obligation to disclose the critical technology. Software code whose revelation could damage national security must be classified per Executive Order 13526 (reference (q)).
 - (3) All other software development that is not a critical technology should consider an open source, collaborative approach that maximizes value through reuse and minimizes maintenance and integration costs. This collaborative approach includes releasing code as OSS whenever practicable on either code.gov or code.mil.
- C. Software, including code fixes and enhancements, developed for systems other than National Security Systems (NSS), should be *open-by-default* and released (under an open source software license) and shall be advertised at either code.gov or code.mil unless any of the following conditions apply:
- (1) The project manager, program manager, or other comparable official has determined in writing that the software constitutes a critical technology that must be protected.
 - (2) The Government lacks the rights to reproduce and release the item, or to authorize others to do so.
 - (3) The public release of the item is restricted by other laws or regulations, such as the Export Administration Regulations or the International Traffic in Arms Regulation, and the item cannot qualify for Distribution Statement A per DoD Instruction 5230.24 (reference (o)).
- D. Software that meets the criteria for Distribution Statements B through F, as defined by DoD Instruction 5230.24 (reference (o)), is considered Controlled Technical Information (CTI) and may not be released as OSS. Program managers should mark such software as Controlled Unclassified Information (CUI), per DoD Instruction 5200.48 (reference (r)). Software that does not meet the criteria for Distribution Statements B through F is not CTI, and should not be treated as such nor marked as CUI.
- E. Components of NSS that are not critical technology, as defined by reference (o), may be released as OSS at the discretion of the program manager or the direction of the relevant Principal Staff Assistant or Component CIO.
- F. Vulnerability information about OSS (including projects maintained by DoD) is unclassified information that shall be handled as Impact Level 2 information in accordance with the DoD Cloud Computing Security Requirements Guide (reference (s)). For DoD-led projects, this information shall be withheld from public release until the vulnerability is analyzed and fixed. Vulnerability information for OSS should be reviewed by the Vulnerabilities Equities Process described in reference (t) before being disclosed to the broader OSS community.
- G. DoD programs releasing code as OSS may use any of the following licenses, which have been shown to be acceptable for DoD use: Apache-2.0, BSD, GPL, LGPL, and MIT

licenses. Component CIOs may grant permission to use other licenses if required. Programs should not develop their own unique license, which creates unnecessary legal complexity. Programs should also consult with a cognizant attorney-advisor about which OSS licenses best meet the Government's needs and do not conflict with applicable statutes, regulations, and policies.

- H. Projects releasing code as open source software to the public or mission partners should publish their intention and ability to accept submissions of proposed enhancements (patches). If the project team accepts submissions from the general public, they must also publish how they will assess, test, and evaluate those submissions for possible incorporation into the managed software baseline.
- I. Whenever a DoD Component first releases a software codebase as OSS or for government-wide re-use (with the exception of basic research funded as part of Budget Activity 1), it must report that to its respective CIO. That CIO must in-turn report it to code.gov. This process enables sharing of such code in compliance with reference (b).
- J. In cases where the management of an existing software codebase is unclear, DoD Components and Component CIOs are authorized and encouraged to make any necessary determinations to resolve ambiguity and release legacy code as OSS where warranted. For example, a CIO may choose to release "orphaned" software that was developed by a program that has since been terminated, but may be useful to other agencies.

4. CONTRIBUTION TO OPEN SOURCE SOFTWARE PROJECTS

- A. Employee participation in OSS projects used by DoD is often in the Government's interest, and is typically a legitimate use of government resources when the Government uses the software in question, either directly, as a component of a larger government system, or as a component of underlying government infrastructure.
 - (1) Government employees may contribute to existing OSS projects (including being the primary maintainer) as part of their official duties, so long as they consult with their supervisor first to ensure a common-sense approach for contributions that preserve Operations Security (OPSEC) and further the Government's interests.
 - (2) Contractors may contribute to OSS projects at government expense when the government task monitor authorizes such activity as being in the Government's interest, subject to the scope and provisions of the contract.
- B. Creating a separate, DoD-specific version of any OSS project, for any reason, increases support risk and should be avoided whenever possible. Creating such a DoD-specific fork creates risk by requiring separate maintenance, reducing access to software capabilities from the OSS community, and may require the Department to assume full management responsibility of the software's source code, thereby increasing costs.
- C. Any improvements or new capabilities added to an OSS project should be contributed back to the upstream open source project via pull/merge request, in accordance with the processes used by that project.