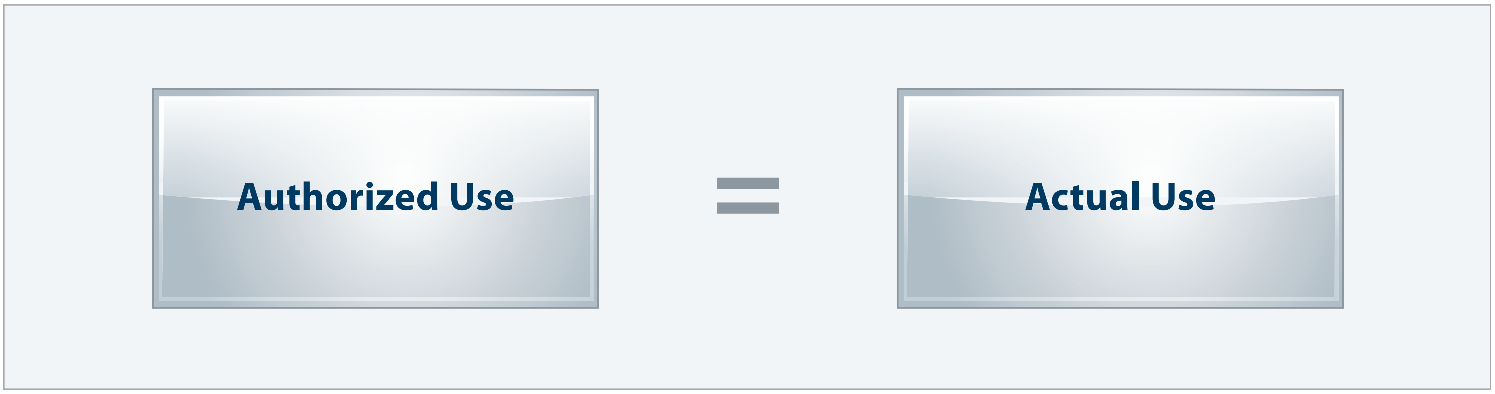
Software Self-Audit Checklist

An Introduction to Software Self-Audits


Authorized Use = Actual Use

A software audit is a defensible comparison of the actual Software Programs, quantities, and uses within an organization measured against the contractually authorized Software Programs, quantities, and uses.

Software audits became an issue in the 1980s and 90s, when Publishers believed that software buyers were utilizing more applications, allowing more users, or utilizing the software other than as licensed and/or purchased.

To avoid having to grant Publishers unlimited access to the customer's systems and data—and the risk of Publishers abusing such a privilege—commercial entities established limits on the rights of a Publisher to perform software audits, as well as on the methods and processes used.

Due to National Security issues, the DoD needs additional protection from outside personnel accessing, viewing, and sharing DoD data as a result of audits.

To provide this protection, the DoD ESI has authored the following language for DoD software license agreements so that the DoD office may perform self-audits:

> **DoD ESI Standard Self-Audit Contract Clause:**
> "Notwithstanding Publisher audit provisions to the contrary, DoD may perform an internal audit of Software use and will use its best efforts to keep full and accurate accounts that may be used to properly ascertain and verify numbers of licenses, users or subscription parameters in use. Upon Publisher written request, DoD may provide audit reports to Publisher from DoD's internal audit records as the sole means of satisfying Publisher's requests for audit."

The self-audit process and scope are pre-defined, with the methods and tools to be used agreed upon in advance within the software license agreement. If over-use or under-use is discovered as a result of the self-audit, remedies are to be implemented *(e.g. payment or reduction in maintenance).*

Standard Approaches to Software Audits

There are three general approaches to software auditing, revolving around who controls the audit:

WHO CONTROLS THE AUDIT?	PUBLISHER	BUYER	THIRD PARTY
GENERAL CONCEPT	Software audit rights were often one-sided in favor of the Publisher, often finding minor/peripheral issues and penalizing the licensee excessively.	Smart buyers have denied Publisher audit rights and third-party audits, and instead agree to perform self-audits on a periodic basis.	Some firms, especially those not in a strong negotiating position, are still able to deny a Publisher audit rights and may agree to an audit by an independent third party.
HOW IS THE AUDIT PERFORMED?	The Publisher audits the licensee's usage with Publisher personnel and tools, reporting findings (often with a bill) to the Licensee.	Self-audit is accomplished by the Licensee utilizing internal personnel to perform the audit under pre-established processes and methods, and report results to their own internal management and the Publisher. Often senior management is required to certify the audit was conducted as established, and that the results are accurate.	A third-party audit is accomplished by hiring an independent company to perform the audit under pre-established processes and methods, and reporting results to the Publisher and User.
		Numerous software tools that search networks and report software instances and usage are available. Some are "certified" by certain Publishers. But that list is small and excludes many well-recognized and tested mainstream products that should give Publishers no cause to dispute their use, accuracy or effectiveness.	Third-party audit rights are defined contractually and should clearly identify the entity performing the audit, which automated tools they will use, the reporting process, the remedy process, who pays for the audit, and whether or not the third-party audit results are binding.
RISKS AND CONCERNS	Several instances where Software is canceled but unintentionally remains on the servers can cause significant non-compliance situations.		
	Some Publishers abuse the situation, charging retail list price license fees, list price maintenance fees, penalties, and interest for all non-compliance situations—including minor or unintended non-compliance.		
	DoD organizations do not want Publishers or third-party personnel accessing sensitive DoD IT Systems, regardless of the firm's reputation and the clearances of their staff. Unmanaged personnel should not be allowed to explore DoD IT systems.		

Typical Results of a Self-Audit

Comparing actual software use to the rights granted by the terms of the contract generally yields one of the following results:

COMMON RESULTS FROM AUDIT	CONSEQUENCES / RAMIFICATIONS
a) Compliance across the entire organization.	None.
b) Software acquired, but not utilized, and with no future plans to utilize.	DoD might be able to save a significant amount of annual maintenance cost, based on the documented under-utilization.
c) Software acquired, but utilized far less than licensed.	
d) Software acquired, but utilized in excess of license grants.	DoD might prevent a complicated situation where the software Publisher could take vigorous actions to charge DoD for improper use. Being proactive, DoD can elect to acquire additional license rights to bring the organization into compliance, or could elect to limit usage within the organization to comply with existing license grants.
e) Software utilized with no record of acquisition or license grants.	DoD needs to determine if license rights exist—from previous organizations or users inherited via reorganization—and then take steps to ensure compliance by acquiring appropriate license rights, or eliminating the software use within the organization.

A Self-Audit Roadmap

1. Establish Contractual Right to Perform Self-Audits

STEP	ACTION	DESCRIPTION
1.1	Contractually limit software use audits to DoD Self-Audit, and establish the procedures in the DoD Software License contract documents.	All DoD purchases of COTS Software should include the contractual term that limits any software audit to a DoD self-audit. This language appears in all DoD ESI BPAs and should be used in any other contract, agreement, order, or other acquisition of COTS Software.

2. Establish a Central Authority and Policy to Ensure Compliance with Software Usage Rights

STEP	ACTION	DESCRIPTION
2.1	Designate a central software asset management (SAM) authority.	The SAM authority is responsible for managing software (and perhaps even all IT Assets) at the appropriate organizational level.
2.2	DoD program offices should proactively detail self-audit issues:	…who specifically will perform the audit …what positions in the DoD organization will be involved …which automated tools will be used …how results will be measured …identify the types of reports and the levels of detail that are needed …what is the reporting process and timeline …what (if any) is the remedy process …who pays for the audit (often varies depending on compliance and/or amount of non-compliance).
2.3	Issue to software users a policy and instructions on restrictions and usage reporting.	Establish and publish written policies defining the approved systems, procedures, and tools for recording license agreements, associated use rights, software received, and software deployed.

3. Establish Four Essential Data Repositories

STEP	ACTION	DESCRIPTION
3.1	Establish data requirements for key repositories.	The central SAM authority should establish the form and data requirements for a **Software License Agreement Repository**, a **Software Receipt Repository**, a **Software Deployment Repository**, and a S**oftware Changes/Modification Repository**.
3.2	Establish the **authoritative repository** of all license grants.	The SAM central authority must maintain central knowledge and an accurate record of license grants and rights of use.

4. Maintain Software Asset Information

STEP	ACTION	DESCRIPTION
4.1	Enter and maintain the **records** and **documentation** of all software license grants via purchases, mergers, etc., establishing the baseline for what software the organization has, and what rights (users, quantities, use restrictions, etc.) accompany the license grants.	The repository should contain copies of all software license agreements, a list of all software actually received, all software deployed, and the associated rights of use (the license grants). This eliminates the extremely difficult and risky approach of a reactive effort to capture data if and when a compliance issue arises.
4.2	At the time an order is placed, the central authority should **record the key data elements** of the license agreement in the DoD approved Software License Repository, a system, tool or database used to retain software license information.	The recommended license data elements to be recorded are as follows: 1. Licensor name and address 2. Licensee name and address 3. Date of license agreement 4. Contract Number and other reference numbers of related documents (Attachments, Price Lists, EULAs, etc.) that specify all Terms and Conditions 5. Type of license (perpetual, term, subscription, etc.) 6. Products or applications licensed, including release level and other identification information 7. Restrictions on use 8. Number and type of users (or processors or other unit of measure) 9. Audit provision (self-audit or other).
4.3	Capture and Record Software Information at the Time of Receipt of the software in the **Software Receipt Repository** section.	The elements recorded should be the same as listed above, along with the date of receipt. Any discrepancies between the license information and the receipt information should be noted and immediately reported to the appropriate contracting officer, DoD POC, and the Licensor POC. The central authority should note corrections as received and follow up to resolve the discrepancies.
		In the spirit of ITIL (Information Technology Infrastructure Library) and IT Service Management, this process should also apply to the receipt of the following: 1. Authorization for additional seats (or other unit of measure) for existing software. 2. Patches, fixes, or other types of software received from the Licensor designed to correct software defects, including any changes to release level or other product identification information. 3. New releases or upgrades received from the Licensor under an existing license agreement including any changes to release level or other product identification information. 4. Negotiated and approved changes to license rights.

4.4	Capture and Record Software Deployment Information at Time of Deployment in a **Software Deployment Repository**.	**Deployment Repository** should include the server name where the software is installed along with the user IDs and the number of user IDs of all authorized and deployed users (or the relevant unit of measure for the number of authorized licenses—e.g., processor names if the license is based on processors).
		As with Software Receipt, all additions, fixes, patches, upgrades, new releases or other changes to the software must be recorded in the Software Deployment Repository once the software is successfully deployed.
4.5	Proactively assist the rest of the organization to ensure compliance, to eliminate duplicate and unneeded licenses (thus minimizing software costs), and to interface with software Publishers in the event of potential non-compliance issues, to include managing any audits.	The software user must ensure they are using the software in compliance with the rights granted via an End User License Agreement (EULA) or other software license agreement executed upon acquisition of software.

5. Execute a DoD Self-Audit

DoD should conduct a self-audit of software use, either as a periodic preventative maintenance approach, as prescribed in the contract, or when a potential non-compliance situation is suspected.

<table>
<tr><th>STEP</th><th>ACTION</th><th>DESCRIPTION</th></tr>
<tr><td>5.1</td><td>Establish a "License Rights" checklist.</td><td>The repository of all license grants is the starting point of the audit. The DoD self-auditors use that data to serve as the baseline set of rights detailing all authorized software use.</td></tr>
<tr><td>5.2</td><td>Execute the audit.</td><td>Using a combination of user interviews and commercial tools that identify software usage on a network, usage details are recorded on a "Software Use Register".</td></tr>
<tr><td>5.3</td><td>Record results on the "Software Audit Findings" document to compare the License Rights to the Software Use Register.</td><td>Results of the self-audit are shared with management. Should any non-compliance issues be detected, decisions addressing how the organization will be brought into compliance are made.</td></tr>
<tr><td colspan="3">Upon review, the DoD Organization might recognize non-compliance situations that can then be addressed in a systematic, negotiated approach with current budget cycles in mind, as opposed to experiencing ad-hoc audits and non-compliance situations where DoD is unable to manage the process within normal budget and program cycles.</td></tr>
<tr><td colspan="3">As a result of proactive self-audit practices, there may be several instances where the DoD Organization determines that they are paying significant fees (especially software maintenance and software support fees) for software no longer being fully used or used at all. In these instances, significant cost savings could more than offset the cost of the organization's software asset management program.</td></tr>
</table>

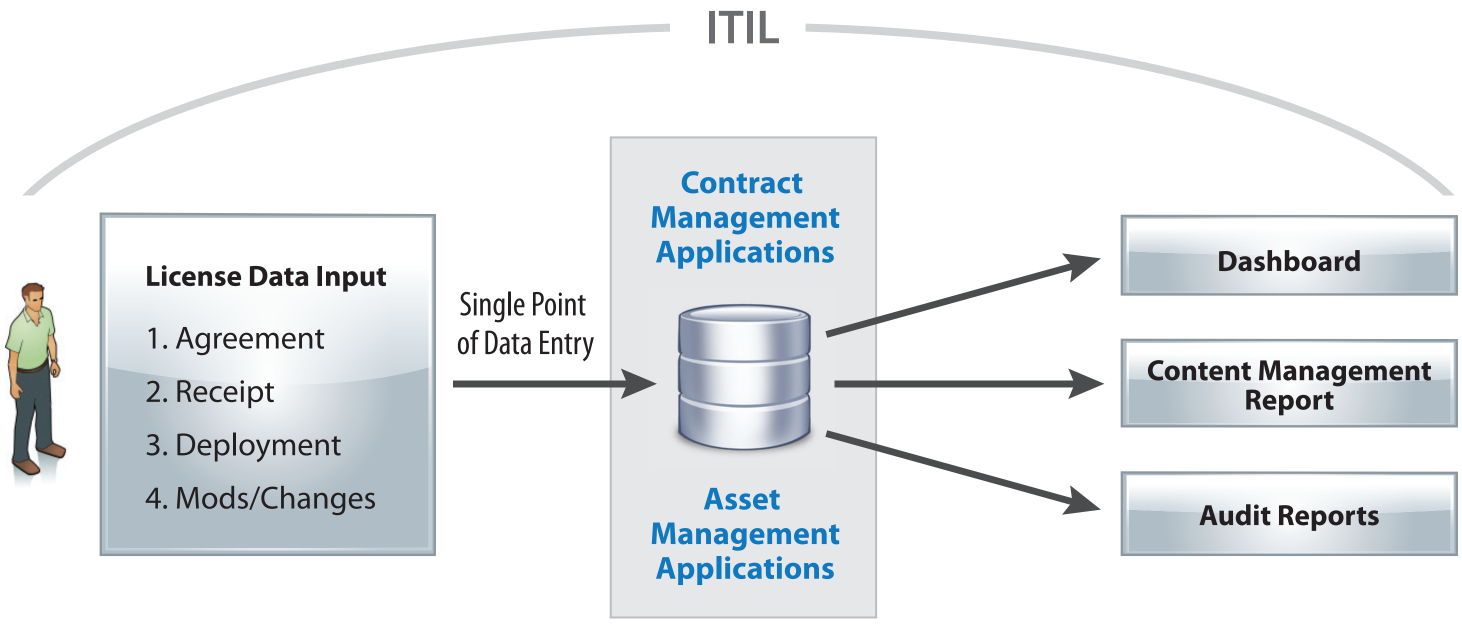
Software Asset Management Tools Overview

Servers, networks, business applications, desktops, and other IT components are often scattered across numerous locations around the world. This makes keeping track of technology assets quite difficult. Compounding the problem is the fact that technology inventory is often managed using manual, paper-intensive processes, which drain resources and are highly prone to inaccuracies and inconsistencies.

With **software asset management (SAM)** tools, organizations can fully automate the process of monitoring and accounting for all their technology assets, allowing for enterprise-wide software audits in a matter of minutes instead of days.

SAM tools provide full insight into how many licenses have been purchased, how many have been allocated and to whom, and how many are unused. This can help licensees avoid purchasing additional licenses they don't need. With automated SAM tools, licensees can also better track software license allocation and better prepare for license renewal negotiations by knowing what needs to be renewed and how many licenses are actually required.

The basic technology components for automated SAM are shown below:


ITIL
License Data Input
1. Agreement
2. Receipt
3. Deployment
4. Mods/Changes
Single Point of Data Entry
Contract Management Applications
Asset Management Applications
Dashboard
Content Management Report
Audit Reports

REPOSITORIES	AGREEMENT	RECEIPT	DEPLOYMENT	CHANGES / MODIFICATIONS
SUMMARY	Captures all appropriate license agreement data.	Captures all license receipt information comparing receipt with license agreement. Any discrepancies should be documented and resolved.	List server name (or other device) and location where software is deployed.	Captures details regarding software updates, patches, fixes, etc.
DATA INPUT ELEMENTS	• Product • Version • Publisher • Vendor • Agreement date • Number of licenses	• Date • Quantity • Etc.	• Date • Quantity • Device • Location of software deployment	• Date (due and actual) • Quantity • Device • Location of software changes

IT Contract Management

1. Organize all software licenses and IT contracts; keep track of terms and conditions.
2. Align IT contracts and software licenses with the assets relating to them.
3. Know when your contracts are up for renewal and maximize your renewal process.
4. Keep track of the vendors you do business with and what you purchase from them.

Conclusion

DoD should ensure that any software audit rights granted as part of an Enterprise Software acquisition are self-audit requirements, and are granted using the unaltered DoD ESI Standard Self-Audit Contract Clause.

Self-Audits are not only a compliance review on behalf of the Software Publisher/Vendor, they are also a valuable internal tool to proactively manage software licenses and costs. Self-Audits are a key component of overall Software Asset Management (SAM) and Information Technology Asset Management (ITAM).