

# Using ISO/IEC 19770-2 Software Identification Tags to Enhance Software Asset Management



## White Paper

Paul 'Doc' Burnham

Paul 'Doc' Burnham has more than 20 years IT experience with an extensive background in Software Asset Management (SAM), software compliance, IT asset procurement, contract and vendor management, and financial analysis. Doc is recognized for his expertise in both the mechanics and the implementation of SAM. His activities within the SAM ecosystem have included development work on ISO/IEC 19770-2 and -3, developing and presenting ITAM- and SAM-related training, authoring white papers, and working jointly with other SAM practitioners, tool vendors and software publishers. Doc is currently the Enterprise Software Asset Management Lead at Textron.

Agnitio Advisors  
1484 Pollard Road, #135  
Los Gatos, CA 95032

Phone: 408-871-0482  
[www.agnitioadvisors.com](http://www.agnitioadvisors.com)

## 2 Using Software Identification Tags For SAM

### Introduction

Most organizations recognize the need to manage their software assets, and those same organizations also recognize that an effective Software Asset Management (SAM) program is not something that just happens. Identifying all software installed in the organization's environment is complex and difficult. This white paper provides an overview of how organizations can use and benefit from software identification tags to enhance their SAM programs.

***Software Asset Management (SAM) is "all the processes and infrastructure necessary for the effective management, control and protection of the software asset within an organization throughout all the stages of its lifecycle." (ITIL Best Practice Guide – Software Asset Management)***

---

### Background

Most organizations understand that software is an expensive and valuable asset. In fact, it is not uncommon for organizations to have more invested in software assets than they do in hardware assets. It is therefore no surprise that organizations want to implement SAM programs in order to more efficiently and effectively manage their software assets.

An effective SAM program enables an organization to answer the following questions:

- What software licenses does it own?
- What software is deployed/installed within its environment?
- What software is actually in use?

The ability to answer these critical questions provides the organization the knowledge it needs to reduce or eliminate various risks and costs, both directly and indirectly associated with SAM. Over-licensing and under-licensing of software are probably the most easily recognized issues, and they are directly associated with SAM. The lesser-known issues that are not usually directly associated with SAM include platform stability, productivity, security and business continuity. Let's briefly review each of these issues.

#### Over-Licensing and Under-Licensing

Without an effective software asset management program in place, the risk of being under-licensed on some applications and over-licensed on other applications is high.

The reduction or elimination of under-licensing software removes or lessens the risk that the organization will face unbudgeted software expenses and penalties related to software licensing non-compliance.

Organizations that reduce or eliminate over-licensing of software reap the cost savings benefit of their actions. Cost savings include the:

- Cancellation of renewals of support for unused software
- Harvesting and reuse of unused licenses

#### Platform Stability\*

When a software asset management program is not effective, there is a high probability that employees are installing and using software that is not company-approved. Before approving software for use within the organization, it should pass rigorous inter-operability and sociability testing with other applications in

---

\* Quoted with permission from Agnitio Advisors, Inc. "Optimizing SAM: Who benefits?"

## 3 Using Software Identification Tags For SAM

the standard operating environment. Since unapproved software has not been tested, it may have negative impacts on critical business applications or network stability.

### Productivity\*

When a software asset management program is not effective, it is very likely that office productivity applications are not kept on a consistent version organization-wide. Version inconsistency creeps in when:

- Employees independently upgrade their software
- Non-centralized purchasing results in new computers with OEM versions of productivity software installed

Having multiple releases of productivity tools in use within the organization can result in some members of the organization being unable to work with files and data from others, because of incompatibilities in file formats produced by different versions of an application. While there are often workarounds for this problem, time is wasted ensuring that files and data are usable.

### Security\*

When a software asset management program is not effective, it is likely that you do not have complete knowledge of all the software installed within your environment. How do you know if the software your end-users are running has been updated with the current security patches? Do you have unknown software installed on corporate systems? If so, that software may include malware, and could easily be compromising network or data security.

### Business Continuity Issues\*

Even if your organization has disaster recovery plans, disaster recovery centers, or backup management plans, these strategies are incomplete without an effective SAM program in place. How do you know what software needs to be installed on computers at a disaster recovery site if you don't know what's installed in your current environment? How do you recover systems from a backup without also addressing which applications are required? Do you have the media and key codes necessary to install software at the disaster recovery site?

## How Software Identification Tags Fit Into SAM

Since so many SAM practices rely on a baseline of accurate software inventory, wouldn't it be nice if each software application came with an identification tag so it could accurately be identified? Of course it would! As part of the journey to that stage, the ISO/IEC 19770-2 standard for software identification tags was developed. This standard outlines how software identification tags should look, and the information they should contain.

***ISO/IEC 19770-2, the standard defining software identification tags, defines a software asset management data standard for software identification tags. Software ID tags provide authoritative identifying information for installed software or other licensable item (such as fonts, or copyrighted papers).***

***A follow-on standard, ISO/IEC 19770-3, the standard defining software entitlement tags, is currently under development. It will outline how software entitlements information can be provided in an automated, electronic fashion to the customer. ISO/IEC 19770-3 is structured in a similar fashion to ISO/IEC 19770-2, and the two standards are designed to be closely inter-related.***

---

The greatest benefit of software identification tags is that they allow an organization to identify what software is installed within its environment. From there, the organization can utilize that knowledge to

## 4 Using Software Identification Tags For SAM

better manage its software assets and facilitate other functions like ensuring platform stability, managing security and planning for disaster recovery/business continuity.

In addition to software identification, ISO 19770-2 provides the ability for the end user organization to add additional information to the software identification tag for its own use, in the 'extended elements' section of the tag. This additional information can further enhance the organization's ability to manage its software assets. Examples of additional information that it may be useful to include are:

- The business unit that purchased the license
- The installation package identifier
- A unique license identifier for a specific site
- For software discovered on a machine, but with no accompanying software identification tag, there is a very good chance the software was installed outside of the normal process.

Today, Adobe is the only major software publisher that has implemented a software identification tag. Adobe implemented its software tag prior to the finalization of the ISO 19770-2 standard, and therefore does not conform to it. However, Adobe's tagging does provide the ability for organizations to accurately identify installed Adobe software. Adobe introduced its software tagging with the CS4 product line.

Adobe's software tag enables organizations to identify an installation of Adobe Acrobat 9 Pro as:

- A stand-alone product
- A trial version
- Part of a Creative Suite package (Design Premium, Design Standard, Web Premium, or Master Collection)

Prior to ISO 19770-2's software identification tag, end-user organizations sometimes created their own marker files to assist them in identifying software. These marker files were simply a primitive version of what is now the official software identification tag. Organizations would include these marker files along with the application software when they repackaging software for deployment internally. As this repackaged software was deployed, so too were the marker files. The use of these marker files provided the organizations with the ability to identify and track their software with a more defined level of accuracy.

Organizations discovered two main issues when using these marker files:

- When they changed auto-discovery tools, they had to modify the new tool to accommodate these files
- Since these files were not standardized, as personnel changed they had to be taught how the files were created and maintained.

As well as these operational difficulties, there is also the issue that marker files are not standardized across companies, meaning each organization must develop and maintain its own tools and practices.

Software identification tags provide all the benefits of marker files, with none of the associated problems.

Because ISO/IEC 19770-2 software identification tags are standardized, organizations can use them to more accurately identify software. As software identification tags are widely taken up, discovery and SAM tools will be upgraded to completely and natively support them.

The table below provides information on how end-user organizations can use ISO/IEC 19770-2 software identification tags today to manage software assets that do not currently include software tags. Note that the standard explicitly defines details that allow a SAM practitioner to automatically identify software identification tags that have been created by an end-user organization versus those created by a software publisher. These provisions mean that as software publishers transition to providing tags themselves, the user-created software tags will not interfere in any way with publisher-created software identification tag files.

## 5 Using Software Identification Tags For SAM

### How software identification files can be used on existing software titles

- A software identification tag (swidtag) is created for each software asset. The software identification tag file is created as defined in the ISO/IEC 19770-2 standard. (Note that this standard is currently in Final Draft International Standard status – contact the convener of ISO/IEC 19770-2, Steve Klos ([stevek@tagvault.org](mailto:stevek@tagvault.org)), for information on how to access this document prior to ISO's publication.
- Each application (including different versions at the first 'dot' level) is assigned a unique swidtag.
- Standalone products use one swidtag.
- Products that are purchased as part of a suite use a swidtag to identify the suite, and individual products within the suite are assigned their own swidtags that reference the suite.
- For stand-alone applications, the swidtag is created by using the product name.
- For applications that were purchased as part of a suite, the swidtag is created by using the product name with the suite name included in parentheses.
- For applications designated for non-production (e.g. test, development, etc.), the swidtag is created as above with "non-prod" used as a suffix.
- To track a Client Access License (CAL), an install package that contains only the software identification tag is created and deployed to the computers that require access, or whose users require access, to the application.

Examples:

Application	Purchased as	Software Tag
Photoshop CS3	Stand-alone	Adobe_Photoshop_CS3.swidtag
Acrobat 8 Professional	Stand-alone	Adobe_Acrobat_8_Pro.swidtag
Illustrator CS3	Stand-alone	Adobe_Illustrator_CS3.swidtag
CS3 Design Standard	Suite	Adobe_CS3_Design_Standard.swidtag Adobe_InDesign_CS3_(CS3_Design_Standard).swidtag Adobe_Photoshop_CS3_(CS3_Design_Standard).swidtag Adobe_Illustrator_CS3_(CS3_Design_Standard).swidtag Adobe_Acrobat_8_Pro_(CS3_Design_Standard).swidtag Adobe_Bridge_CS3_(CS3_Design_Standard).swidtag Adobe_Device_Central_CS3_(CS3_Design_Standard).swidtag Adobe_Version_Cue_CS3_(CS3_Design_Standard).swidtag
MSSQL Server 2005- SE	Stand-alone	Microsoft_SQL_2005_SE.swidtag
MSSQL Server 2005- EE	Stand-alone	Microsoft_SQL_2005_EE.swidtag
MSSQL 2005 CAL – User	Stand-alone	Microsoft_SQL_2005_User.swidtag
MSSQL 2005 CAL – Device	Stand-alone	Microsoft_SQL_2005_Device.swidtag
Oracle 10g DB – SE	Stand-alone	Oracle_DB_10g_SE.swidtag
Oracle 10g DB – EE	Stand-alone	Oracle_DB_10g_EE.swidtag
Oracle 10g DB – EE w/options	Stand-alone	Oracle_DB_10g_EE_w/options.swidtag
MSSQL 2005 Server - SE	Non-prod	Microsoft_SQL_2005_SE_non-prod.swidtag
Oracle 10g DB – EE	Non-prod	Oracle_DB_10g_EE_non-prod.swidtag
Oracle 10g DB – EE w/options	Non-prod	Oracle_DB_10g_EE_w/options_non-prod.swidtag

### How Organizations Can Use Software Identification Tags To Enhance Their SAM Programs

## 6 Using Software Identification Tags For SAM

Software identification tags support organizations in collating and maintaining accurate software inventory records, which are critical to many aspects of SAM programs. Let's first understand the value of software identification tags to inventory collection, and then look at the other aspects of the SAM program that are enhanced by the availability of a complete, current and accurate software inventory.

### How Software Tags Benefit a Software Inventory

Software identification tags enhance an organization's ability to locate installations of software and to accurately identify them. Without software identification tags, it is hard to reconcile software installations against their purchase and license records, because of the many different licensing options available.

Some organizations currently use automated discovery tools that rely on signature files for identification. It's critical to know that changes to an installed application, such as the addition of a service pack or installation of a patch for a security flaw, change the signature of the installed application. For example: if a service pack is added to Microsoft Office 2007 Pro, the signature of that application will change. Therefore the signature file to identify it will also have to change.

The signature file can be updated:

- By the tool publisher (usually monthly or quarterly, to organizations with current support or maintenance agreements only)
- By your organization, if your discovery tool allows you to make updates to its signature library

Software tags offer a simpler and more robust solution for identifying software than the use of signature files. ISO/IEC 19770-2 software identification tags are standardized and therefore can be utilized by the various auto-discovery tools to identify software, regardless of the platform or publisher.

### Reconciling Licenses to Determine Compliance

Once software is accurately identified, organizations need to reconcile it with the correct entitlement. This is particularly important for applications that can be purchased under many different licensing conditions. For example:

- Microsoft Access 2007 can be purchased as a stand-alone product or as part of an Office suite (Professional, Enterprise, Enterprise Plus, Ultimate), or installed as a trial version
- Microsoft SQL 2008 can be purchased in different editions (Express, Standard, Enterprise, Workgroup, Web, Developer), or installed as a trial version
- Adobe Illustrator CS4 can be purchased as a stand-alone product or as part of a Creative Suite package (Design Premium, Design Standard, Web Premium, Production Premium, Master Collection), or installed as a trial version

Without software identification tags to correctly identify software installations, this reconciliation process has to be completed manually using purchasing records, licensing details from software vendors and 'best guesses'. With accurate software inventory, it's possible to automate at least some license compliance checking, vastly simplifying and reducing the cost of the compliance effort.

### Financial Decision-Making

Accurate inventory information enables financial decisions to be made, including:

- The most advantageous license model to use for future license purchases. This is particularly important for applications that can be purchased by various license models (by user, by device, by processor, etc.)
- Whether to continue maintenance and support on an application
- Whether to purchase new or upgrade licenses for an application

## 7 Using Software Identification Tags For SAM

### Platform Stability

Determining the variety of software installed within the environment, including the number of versions of particular software, is particularly important if your organization is required to test new critical applications against all known applications within the environment. Interoperability and sociability testing answers questions such as: Do multiple VPN clients cause any issues? Do multiple versions of Internet Explorer cause issues? Are there rogue installs of an application happening?

### Security Decision-Making Based On Risk Assessment

Accurate software inventory enables you to determine if there is software with known security risks installed within your environment. It answers questions such as: Do all Windows XP installations have all the critical patches installed? Are there unauthorized (or unwanted) software applications such as peer-to-peer file sharing or hacking tools installed anywhere on the network? Armed with accurate information, you can make better decisions, and ensure that your security policies are enforced.

### Disaster Recovery/Business Continuity Planning and Operations

An accurate software inventory allows you to determine the locations of business-critical software installations. In turn you can ensure their data is being fully backed up, and ensure that both the software and the data are available in a disaster situation or crisis.

## Calculating the Financial Benefit of Software Identification Tags

It has been proven over and over that an effective SAM program will return an organization various benefits including cost savings and cost avoidance related to:

- Better negotiating positions
- Improved purchasing arrangements
- Improved infrastructure planning
- Reduced risks arising from mergers-acquisition-divestures
- Reduced costs related to license non-compliance

Which of these financial benefits relate directly to using software identification tags? Since software identification tags enable the organization to identify installed software faster and more accurately, there is an immediate financial benefit in the reduced work hours and costs associated with software inventory and license reconciliation.

While analyses such as this can result in 'hard dollar costs' that are relatively easy to quantify, estimating 'soft dollar costs' from outcomes such as reduced risks remains subjective. Here are some benefits of SAM to consider when performing your financial analysis:

#### Easier to calculate

- Reduced maintenance and support costs on applications
- The difference between the initial offer and negotiated cost of an application
- The value of the additional terms and conditions that were negotiated at reduced or no cost
- Reclaiming (harvesting) unused software licenses that are redistributed versus purchasing new licenses
- Costs avoided by verifying that invoices match contract requirements

#### More difficult to calculate

- Costs avoided by not having to purchase over-deployed software as a result of an external audit
- Cost of a damaged organization reputation due to reported copyright violations
- Work hours and cost avoided by preventing software licensing problems related to merger/acquisition/divestiture activities
- Work hours and cost avoided by preventing software-related issues during disaster recovery operations

## 8 Using Software Identification Tags For SAM

Here are some real life examples of actual cost savings or cost avoidance where software tags were used:

- Example 1
- Discovered: Microsoft SQL Enterprise was deployed to a large number of machines
  - Issue: Microsoft SQL Standard should have been deployed
  - Results: Microsoft SQL Enterprise was removed and replaced with Microsoft SQL Standard – avoiding a potential \$100,000+ licensing issue if audited
- Example 2
- Discovered: Microsoft SQL Standard was highly deployed within a large server farm
  - Issue: Microsoft SQL Enterprise should have been deployed
  - Results: Microsoft SQL Standard was removed and replaced with Microsoft SQL Enterprise – avoiding a potential \$750,000+ licensing issue if audited
- Example 3
- Discovered: Oracle DB Enterprise w/options – non-prod was deployed to machines
  - Issue: Oracle DB Standard should have been deployed
  - Results: Oracle DB Enterprise w/options – non-prod was removed and replaced with Oracle DB Standard – avoiding a potential \$150,000+ licensing issue if audited
- Example 4
- Discovered: a solution provider installed Microsoft SQL Enterprise as the back-end database for the company's HR system
  - Issue: proper licensing could not be verified
  - Results: proper Microsoft SQL Enterprise licenses were obtained – avoiding a potential \$65,000 licensing issue if audited
- Example 5
- Discovered: a solution provider installed Adobe Acrobat as part of their \$65,000 document management system solution
  - Issue: the solution provider could not provide proof of licensing as per contract for the necessary Adobe Acrobat licenses
  - Results: solution provider purchased required licenses – avoiding a potential \$250,000+ licensing issue if audited

### Conclusion

As organizations' SAM programs mature, the organizational knowledge and understanding related to SAM will also mature. Software identification tags, and the use of them by organizations, are part of this growth to maturity. The sooner organizations take advantage of software identification tags, the sooner they will realize the benefits they can provide, especially in the reduction of time needed to identify and reconcile software licenses to deployed software. As organizations embrace the use of software identification tags, they should request or demand that software publishers offer software identification tags as outlined in ISO/IEC 19770-2.