



# Standard Published: What Now?

## *How Will the ISO/IEC 19770-2 Standard for Software Tagging Influence Your Organization's Decisions?*

**On November 15, 2009, the ISO/IEC 19770-2 standard was finally published. It is available to the public, and now it is time to make final decisions as to how the passage of this standard will influence our organizations and how we should react to it.**

### **What now?**

The driving forces of the ISO/IEC 19770-2 standard are the people that negotiate contracts. This specification is designed to make their lives easier. As Software Asset Managers meet the daily problems with identifying what they have on their machines in their environments, they will surely put pressure on software vendors to adhere to this new standard.

Why not add the clause into the contract that requires adherence to the standard? Why not require that the tag is verified by an independent third party organization (like TagVault.org)? If this is made a policy and best practice, the pain of software identification will be more bearable. The advantages of software tagging for software asset managers will be very significant. Significant external pressure from the consumer on software vendors to tag their software as a standard practice can expedite compliance.

Another adoption power lies in the public sector. Having the software tag seems to have very strong potential of becoming a common requirement. I accept the argument that some vendors will choose another market instead of investing in the tagging process for their development environments. However, I doubt that they will form a majority. I am afraid there is so much money behind the public tenders that the vendors that give up will soon be marginalised. For that very same reason, the software identification tags (SWID tags) will be the more popular choice. After all, they are in business to make money.

The Software Identification Tag is not only for big, commercial software vendors to implement. Though Software Asset Management is a financial domain, not only commercially sold software products are available and used within organizations today. For this reason, the ISO/IEC 19770-2 standard got the attention of the Free and Open Source Software (FOSS) community, and now we have it

represented in WG21. We are working together on making future standards more aligned with the FOSS community's needs.

The unique nature of this specification is not only in its quality and focus on Software Asset Management issues. This directive provides comprehensive descriptions of how the tag may and should be created, modified and used. There are 3 top-level sections in the standard that explain what tasks, issues and procedures are related to the software tag implementation in every stage of the SWID tag lifecycle.

In addition, if the standard is just a publication, it may become just another written specification and not utilized. This is however not the case here. There are several projects encompassing this idea that provide much more than just the specification. Services will be provided that facilitate creation and management of the SWID tags. For example, I can imagine the development team in a big software vendor organization that would fill the file property with information about the software title without even knowing what software product or product suite it belongs to. Even if they know this, they may not know what the final product name is. SWID tag moves the responsibility of tagging the software products to the people that actually know what software product they are tagging. It now becomes the responsibility of the organization that sells the software application and not that of the software vendor. Of course tag creation may be connected to the actual software development process, but there is no such requirement at this time.

So, will the ISO/IEC 19770-2 be adapted everywhere? Is there a need to have all vendors tagging their software? Well, if you are considering a software purchase, you would want YOUR vendors to tag their software would you not?

### **Software ID tags vs. file headers**

There were many attempts to organize information about installed software in the past. We had file headers, add/remove programs information, package managers, etc. The main issue with them was that they were platform specific and technology focused. Now we have ISO/IEC 19770-2 that is platform and vendor independent. After all, the main goal of

standardization is to provide common solutions between different vendors and across multiple platforms.

The conversation about the possibility of using file properties for tagging purposes was curbed quickly, as files do not necessarily represent their respective software packages anymore. This is not the only problem. Having multiple executable files for one software package and the differentiation of packages based on secondary files (like dlls) causes confusion. How about software that is installed as just a registry entry that allows a user to access a server? This may be considered a fully installed software package as well and one that needs a license to be reconciled. How would you tag this kind of software with a file property when the file does not exist?

Software identification tags also provide the Software Asset Manager with the ability to manage product suites. This is something that already exists partially in some package management systems. However, SWID tags provide logic to quickly indicate which product is part of which suite, something that is difficult to determine currently. This is something that Software Asset Managers would like to see for every software suite they use within the organization.

Another strong point of SWID tags is that they provide the information about not only the files that belong to the tagged product, but virtually any footprint the product leaves on the machine. That allows the user to identify the related files that were previously unknown or suspicious.

### Legacy software

The ISO/IEC 19770-2 standard does not leave the issue of legacy software out in the cold. If the software product does not have a tag created by the manufacturer, it can be easily added by the reseller. It can be also added by the service provider. How about adding this requirement to an SLA (Service Level Agreement)?

Yes, I want to have the software installed, but I want it to be installed so that I am able to see what is actually installed, without the need to look for the information somewhere in the files' metadata. I want this information in the place I can find easily. Some may argue that this is what is provided by "Add/remove programs" in Windows or package managers under UNIX systems. The issue there is however, that they must be created and provided properly by the manufacturer, who may be not interested in doing that.

With the SWID tag, you get a unified way of creating this information and also services that will help you provided by the third party tag registration organization – TagVault.org. Now, let's go a step further. Let us add the tag to the deployment package. Let us add it to the installation instruction. If I want it, I can have it.

### Not executable software

The software identification tag is designed to be used with all types of software. It means that it is to be used not only with executable software, but also with data software like fonts or



documents. It may need some further work to facilitate usage of the tags for the data software, but it was not forgotten in the standard.

### Tag protection

The software tagging standards were not created to provide the ultimate antipiracy framework. They also do not resolve the problem of a software audit as of yet. They are designed to be an information resource for the Software Asset Manager. They are created to help organizations with compliance adherence. If someone wants to hide an application's existence, there are ways to do it. Even if the audit is based on file names, they can be modified if there is not enough protection and security within the organization. It can be done even by a child trying to hide a favourite computer game on dad's work computer.

Trying to protect the information about the software installed on a device is a different subject. First, let us determine a consistent, standardized way of identifying the software, then; let's think about how to protect it.

What if the tag is hacked or deleted? Well, first of all, the software ID tags are not designed to provide the evidence of the software's existence or non-existence. The tag is the structure providing information about what the installed software is. Note that only 5 of the 100 pages in the standard define the obligatory requirements. If you meet them, you will have the software identified, which is the foundation of what the software identification tag is for. However, if you follow all the guidance of the standard, you will also employ the following security mechanisms that are within the standard as well:

- Signature – Every element of the tag can be signed. You can then easily identify if someone edits the signed elements.
- Self healing – Every time the software is run, it can recreate the tag if it is missing or corrupted.
- Validation – SWID tags contains information about an executable that can check if the tag is correct or not.



- Platform security – Platforms protecting the tags from unauthorised access. There is a chance that the OS vendors may be interested in requiring the software to have a registered SWID tag to start the application.

The tag validation information is optional, but there is a way to check whether or not the tag contains proper information about the software installed. The process of such validation uses one of the tag's elements (<validation>). The information in this element points to an executable that checks if the tag is valid. This executable would need to be created by the software vendor and it is optional.

Talking about protection, there are already mechanisms in place for content protection. They work for music, video and document files. Does it make sense to use the tags that are not protected? First of all, if there is only a slight chance it will be corrupted; there is no need to protect it. Second, you are able to protect the tag files using the existing mechanisms in operating systems. Third, I am pretty sure that the manufacturers do not need the standard to provide the protection mechanisms for the content they want to protect. Nevertheless, when the tags are implemented and accepted, there will be a time when the tags protection mechanisms will need standardization. In fact, the initial research on this subject is being conducted currently, and if you would like to share

your knowledge regarding this domain, you can contact me through the ITAK magazine at [ITAK@iaitam.org](mailto:ITAK@iaitam.org).

A Software Identification Tag is something that is useful and not just as an informational descriptor about the software product. This is not the only component of the ISO/IEC 19770-2 standard that provides the value. Implementation descriptions and the supporting services provide the power to make the standard truly useful.

Now the ball is in your court – the consumer - and we are looking forward to any feedback about how future standardization in this field should appear.

**Krzystof (Chris) Baczewicz**

*ISO/IEC editor of the ISO/IEC 19770-2  
and ISO/IEC 19770-3 and IT Standards  
Support Department Manager  
Eracent Inc.*

