INTERNATIONAL STANDARD

ISO/IEC 19770-2

Second edition
2015-10-01

Corrected version
2017-02

Information technology — Software asset management —

Part 2: Software identification tag

Technologies de l'information — Gestion de biens de logiciel —

Partie 2: Étiquette d'identification du logiciel

Reference number
ISO/IEC 19770-2:2015(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

This second edition cancels and replaces the first edition (ISO/IEC 19770-2:2009), which has been technically revised.

This corrected version of ISO/IEC 19770-2 incorporates the following corrections plus other minor editorial modifications:

— two subclauses have been added to 8.4; and

— the schema for the BaseElement type has been replaced in Annex B.

ISO/IEC 19770 consists of the following parts, under the general title *Information technology — Software asset management*:

— *Part 1: Processes and tiered assessment of conformance*

— *Part 2: Software identification tag*

— *Part 5: Overview and vocabulary*

The following parts are under preparation:

— *Part 3: Software entitlement schema*

— *Part 4: Resource Utilization Measurement (RUM)*

— *Part 7: Tag management*

The following part is planned:

— *Part 22: Guidance for the use of* ISO/IEC 19770-2 *Software Identification Tag information in Cyber Security*

Introduction

Overview

International Standards in the ISO/IEC 19770 family of standards for Information Technology (IT) asset management (ITAM) address both the processes and technology for managing software, hardware, and related IT assets. Because IT is an essential enabler for almost all activity in today's world, these standards must integrate tightly into all of IT. For example, software identification (SWID) tags have the capacity to assist in other management functions outside the scope of financial-focused or compliance-focused ITAM processes. From a technology perspective, ITAM standards for information structures provide not only the data interoperability of software management data, but also provide the basis for many related benefits such as more effective security in the management of software. ITAM standards for information structures also facilitate significant automation of IT functionality, such as improved authentication of software and automated linking to identify vulnerability information for more automated exposure identification and mitigation.

Purpose of this part of ISO/IEC 19770

This part of ISO/IEC 19770 provides an International Standard for software identification tags. The software identification tag is a standardized data structure containing software identification information about a software product that supports new and automated management functions. Product information provided in the software identification tag structure will often be provided in an XML data file, but the same SWID tag product information may be accessible through other means depending on the computing device being managed.

SWID tags are created by a SWID tag producer, for example a software creator who develops and distributes software or a tool and/or service provider. SWID tag data is utilized by SWID tag consumers, for example a discovery tool or service that collects information from a computing device for a variety of purposes such as license compliance, software security, or logistics operations. Providing authoritative and detailed software identification information makes the management of software less expensive and provides support for significantly more automation for IT processes in the security, compliance, and logistics areas.

This part of ISO/IEC 19770 has been developed to facilitate automation of IT processes through the use of software identification tags and for applications which use those tags, for the purposes of security, compliance, and logistics automation. This part of ISO/IEC 19770 includes information which facilitates human intelligibility (such as edition and colloquial version name), but it is unrealistic to expect to create, manage, and use software identification tags without the use of automated capabilities built into specialist or generalist tools. The extent to which such capabilities are provided by specialist commercial products, open-source-type products, or platforms themselves, will depend on market developments over time.

This part of ISO/IEC 19770 supports software asset management processes as defined in ISO/IEC 19770-1. This part of ISO/IEC 19770 is also designed to work together with ISO/IEC 19770-3 which will provide an International Standard for software entitlement schema.

Software identification tags will benefit all stakeholders involved in the creation, licensing, distribution, releasing, installation, and on-going management of software. Key benefits associated with software identification tags include the following.

a) The ability to consistently and authoritatively identify software products that need to be managed for any purpose, such as for licensing, security, logistics, or for the specification of dependencies. Software identification tags provide the meta-data necessary to support more accurate identification than other software identification techniques.

b) The ability to identify groups or suites of software products in the same way as individual software products, enabling entire groups or suites of software products to be managed with the same flexibility as individual products.

c) The ability to automatically relate installed software with other information such as patch installations, configuration issues, or other vulnerabilities.

d) Facilitate interoperability of software information between different software creators, different software platforms, different IT management tools, and within software creator organizations, as well as between SWID tag producers and SWID tag consumers.

e) Facilitate automated approaches to license compliance, using information both from the software identification tag and from the software entitlement schema as specified in ISO/IEC 19770-3.

f) Provide a comprehensive information structure of the structural footprint of products, for example the list of software components of files and system settings associated with a product to identify if files have been modified.

g) Provide a comprehensive information structure that identifies different entities, including software creators, software licensors, packagers, distributors external to the software consumer, as well as various entities within the software consumer, associated with the installation and management of the product on an on-going basis.

h) Through the optional use of digital signatures by organizations creating software identification tags, the ability to validate that information is authoritative and has not been maliciously tampered with.

i) The opportunity for entities other than original software creators (e.g. independent providers or in-house personnel) to create software identification tags for legacy software, and for software from software creators who do not provide software identification tags themselves.

This part of ISO/IEC 19770 is divided into the following clauses and annexes:

— Clause 1 defines the scope;

— Clause 2 describes the normative references;

— Clause 3 describes the terms, definitions, and abbreviated terms used in this part of ISO/IEC 19770;

— Clause 4 defines conformance;

— Clause 5 provides interoperability guidance;

— Clause 6 describes the implementation of software identification tagging processes;

— Clause 7 contains platform implementation requirements and guidance;

— Clause 8 describes the elements of the tag;

— Annex A contains information on why the changes to the SWID tag schema are necessary;

— Annex B contains the XML schema document for the tag;

— Annex C provides a UML diagram of the SWID tag schema;

— Annex D provides sample tags.

Information technology — Software asset management —

Part 2:
Software identification tag

1 Scope

This part of ISO/IEC 19770 establishes specifications for tagging software to optimize its identification and management.

This part of ISO/IEC 19770 applies to the following.

a) Tag producers: these organizations and/or tools create software identification (SWID) tags for use by others in the market. A tag producer may be part of the software creator organization, the software licensor organization, or be a third-party organization. These organizations and/or tools can broadly be broken down into the following categories.

   1) Platform providers: entities responsible for the computer or hardware device and/or associated operating system, virtual environment, or application platform, on which software may be installed or run. Platform providers which support this part of ISO/IEC 19770 may additionally provide tag management capabilities at the level of the platform or operating system.

   2) Software providers: entities that create, license, or distribute software. For example, software creators, independent software developers, consultants, and repackagers of previously manufactured software. Software creators may also be in-house software developers.

   3) Tag tool providers: entities that provide tools to create software identification tags. For example, tools within development environments that generate software identification tags, or installation tools that may create tags on behalf of the installation process, and/or desktop management tools that may create tags for installed software that did not originally have a software identification tag.

b) Tag consumers: these tools and/or organizations utilize information from SWID tags and are typically broken down into the following two major categories:

   1) software consumers: entities that purchase, install, and/or otherwise consume software;

   2) IT discovery and processing tool providers: entities that provide tools to collect, store, and process software identification tags. These tools may be targeted at a variety of different market segments, including software security, compliance, and logistics.

This part of ISO/IEC 19770 does not prescribe Information Technology Asset Management (ITAM) or other IT-related processes required for reconciliation of software entitlements with software identification tags or other IT requirements.

This part of ISO/IEC 19770 does not specify product activation or launch controls.

This part of ISO/IEC 19770 is not intended to conflict either with any organization's policies, procedures or standards or with any national or international laws and regulations.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19770–5, *Information technology — Software asset management — Part 5: Overview and vocabulary*

IEEE 1003.1:2013, *Standard for Information Technology — Portable Operating System Interface (POSIX(R))*

W3C Recommendation, *XML Schema Part 2: Datatypes (Second Edition)*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19770-5 and the following apply.

3.1.1
patch
software component that, when installed, directly modifies files or device settings related to a different software component without changing the version number or release details for the related software component

3.1.2
platform provider
organization responsible for the platform

Note 1 to entry: The platform provider is typically the vendor of the relevant operating system, virtual environment, or application platform.

3.1.3
tagId
globally unique value that shall be globally unique for every SWID tag created

Note 1 to entry: Globally unique values may use a 16 byte GUID, or other globally unique value as defined by the tag creator.

3.2 Abbreviated terms

API	application programming interface
GUID	globally unique identifier
IETF	Internet Engineering Task Force
MD5	message digest 5
regid	registration identifier
RPC	remote procedure call
SAM	software asset management
SHA	secure hash algorithm
SWID	software identification, or software identification tag
URI	uniform resource identifier
URL	uniform resource locator

VAR value added reseller

W3C World Wide Web Consortium

XML extensible markup language

XSD XML schema definition

4 Conformance

4.1 SWID tag conformance

A software identification tag is in conformance with this part of ISO/IEC 19770 if the tag data structure obeys all normative constraints specified in this part of ISO/IEC 19770.

4.2 Application conformance

Application conformance incorporates both syntax and semantics.

— A conforming tag consumer shall not reject any conforming SWID tag.

— A conforming tag producer shall be able to produce SWID tags conforming to this part of ISO/IEC 19770.

— A conforming tag consumer shall treat the information in SWID tag in a manner consistent with the semantic definitions given in this part of ISO/IEC 19770. An application's intended behavior need not require that application to process all of the information in a SWID tag. However, the information that it does process shall be processed in a manner that is consistent with the semantic definitions given in this part of ISO/IEC 19770.

— A conforming tag consumer shall, when necessary, be able to identify the version of the XML schema (XSD) used for a SWID tag and process information provided in older versions of SWID tags in a manner that is consistent with that version of the XSD.

4.3 Platform conformance

A platform is in conformance with this part of ISO/IEC 19770 if it provides a programmatic interface to add, retrieve, enumerate, and remove SWID tag data and/or if it provides support for SWID tags to be stored on and retrieved from a file storage environment on a specified device.

5 Interoperability guidance

5.1 Overview

It is critical that SWID tag producers create SWID tag data structures in a manner so that their tags can be consumed and used by tools and users in a consistent fashion and so that SWID tag consumers understand exactly how they should interpret the relationships defined within the SWID tags. This requires that tags be created in a way that is interoperable with tools and users requirements.

This Clause provides details on how SWID tags are created, what the various relationships defined in the SWID tag mean, and how that information can be interpreted by tools and users.

5.2 SWID tag modification

All SWID tags (regardless if they are primary or supplemental tags) shall only be modified by the organization that initially created the tag; this is to ensure that data, especially digitally signed data, is not modified in any way that the tag creator is not directly responsible. There are many instances

when additional data needs to be associated with an existing SWID tag, so the SWID structure allows any organization to create their own supplemental SWID tag that references another SWID tag (either primary or supplemental SWID tags can be referenced). This allows, for example, a software purchasing organization to associate specific licensing data for a user or device to a software title.

In many instances, the SWID tag will be created and maintained by the software developer. However, in cases where software is discovered that does not have a SWID tag, there will be instances where a discovery tool identifies software on a device and creates a SWID tag for that software. In either case, the entity that creates the initial SWID tag is creating a primary SWID tag and the data provided in the tag shall not be modified by any organization other than the organization specified as the "tagCreator" role of the Entity data attribute (see 8.5.2). For an example of a tag created by a discovery tool, see D.4.

There are many instances where additional information needs to be associated with an existing primary SWID tag. Since SWID tags may not be modified by anyone other than the Entity identified with the tagCreator role and securely signing a SWID tag update in the field may be difficult, supplemental tags are utilized to provide additional data. Supplemental tags may be used to provide details such as software activation information, or specific customer related information such as Information Technology Infrastructure Library (ITIL)-based release testing and rollout details for a specific software product.

5.3 SWID tag relationships

5.3.1 Overview

SWID tags are used to represent the software elements that are part of a software product. There may be multiple different SWID tags that define a software product, the various components that make up the software product, as well as which patches are related to the software product. The relationships between these multiple SWID tags are defined through SWID links that are part of the SWID data structure.

There are three special types of relationships that are uniquely defined in SWID tags: SoftwareIdentity attributes that is for pre-installation data; a software patch and for supplemental tag data.

5.3.2 Pre-installation data attribute

When software is distributed from a software publisher, it is typically provided in a "pre-installation" structure and includes an installation script. This type of distribution may be provided on removable media or through downloaded file. In these instances, there are often times a tag consumer will want to know details about the software that is available to install. SWID tags identifying pre-installation distributions of software can be provided and are identified if the attribute corpus is set to true (see 8.5.1).

5.3.3 SWID patch attribute

When a patch is being identified by a SWID tag, it shall include the attribute "patch" with the value being set to true (see 8.5.1).

SWID tags included with patches shall include a link to the product or products (see 8.5.4) it patches as well as links to any other patches to which the current patch may have a relationship. The relationships (see 8.6.7) that may be used by a patch are as follows:

— patches – every patch with a SWID tag must include a Link to the product(s) it patches and the link must use the rel value of "patches";

— requires – the new patch requires that the earlier patch be installed first. In this case, the new patch may only be installable if the earlier patch is installed;

— supersedes – the new patch includes all files from an earlier patch. The new patch (that supersedes the older patch) may be installed on its own, or after the earlier patch and either installation path will result in the same resulting state of the software product that is being patched.

Patches that do not have either link with a relationship of supersedes or requires may be installed independently of each other and in any order. Figure 1 provides an example of how the SWID tags for patches work.

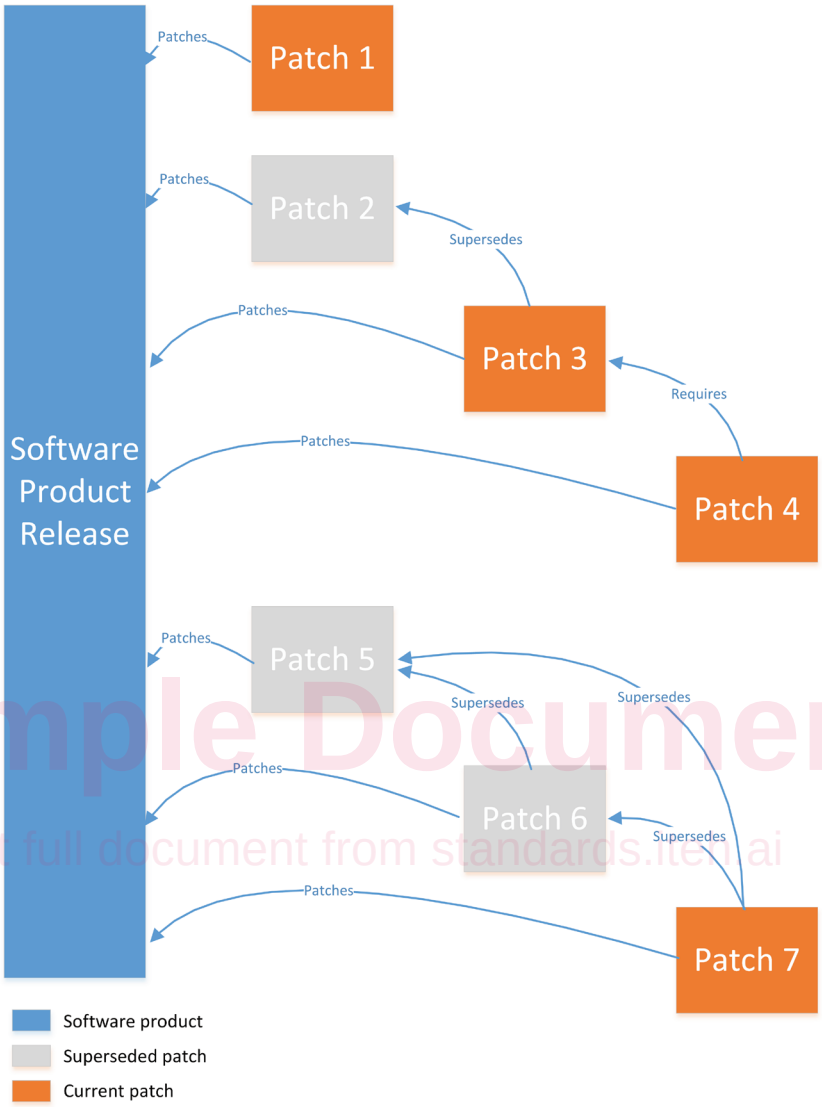

Figure 1 — Representation of patch and product relationships

5.3.4 SWID supplemental attribute

Supplemental tag data is data that is directly associated with a specific software product's primary tag but, for various reasons, the data included in the supplemental tag is not included in the primary SWID tag. SWID tag data may only be modified by the tagCreator; in other words, if a software creator provides a primary SWID tag for their product, the software consumer who installs and manages that software is not allowed to modify any data in the primary SWID tag. In this case, the software tag consumer can create a supplemental tag that provides specific details for the primary SWID tag they are referencing and they will set the attribute "supplemental" to the value of true (see 8.5.1). This supplemental tag can then be deployed with the installation of the software, or added after the fact as part of a device management process, or a software activation process.

Supplemental tags may also be provided by the tag creator to add additional information related to a specific installation of a software title.

6 Implementation of software identification tagging processes

6.1 General requirements and guidance

6.1.1 XML and XSD

The software identification tag file shall be defined as an XML data structure. The XML schema definition (XSD) as specified in this revision may be downloaded from

http://standards.iso.org/iso/19770/-2/2015/schema.xsd

In anticipation of possible defects which will be identified after publication, an alternative download location always with the latest defect-corrected version is

http://standards.iso.org/iso/19770/-2/2015-current/schema.xsd

6.1.2 SWID tags based on earlier revisions of this part of ISO/IEC 19770

A SWID tag (or tags) installed as part of a specific software product are assumed to continue to exist until the software components installed as part of that release are modified in some way. Tag consumers should be aware that earlier versions of the XML schema definition (XSD) may exist for older software products and that the older schema can be located at standards.iso.org.

Tag producers and tag consumers who are creating tags based on this revision are not required to also provide a SWID tag based on any older revisions.

6.1.3 SWID tag installation and removal

In instances where a software product is installed on a device, a software licensor conforming to this standard will ensure that a primary SWID tag is included on the installation media and installed at the same time the software is installed.

When software is uninstalled, or changed to a different release, the old SWID tags shall be removed from the device.

In instances where a patch is installed, the patch will include a patch SWID tag that will be installed when the patch is installed, and in most cases should be removed when either the patch is uninstalled, or when the product is uninstalled, or changed to a different release. The determination if a patch tag is to be removed, or not, is based on additional data provided in the ownership attribute (8.6.3).

Supplemental tags provided by the software publisher (which may be used to identify relationships between software products) shall be managed in a manner similar to primary and patch SWID tags such that the supplemental tags should be removed from a device when the software product is uninstalled.

As noted in 6.1.4, SWID tags reside in the same directory tree as the applications installation directory tree. It is expected that if an application directory tree is deleted when an application is uninstalled, that the SWID tags associated with that application (including primary, supplemental, and patch tags) are deleted as well.

6.1.4 SWID data storage and transmission

On devices with a file system, but no API defined to retrieve SWID tags, the SWID tag data shall be stored in an XML file and shall be located on a device's file system in a sub-directory named "swidtag" (all lower case) that is located in the same file directory or sub-directory of the install location of the software component with which they are installed. It is recommended, but not required, that the swidtag directory is located at the top of the application installation directory tree. Any payload information provided shall reference files using a relative path of the location where the SWID tag is stored.

On devices that do not have a file system, the SWID tag data shall be stored in a data storage location defined and managed by the platform provider for that device. SWID tag data stored in this manner may be transmitted using an alternative format in specific use cases. This may include a computing device that utilizes an API to store and retrieve SWID tag information and/or data-interchange formats other than XML.

On devices that utilize both a file system for software installation as well as API access to the SWID tag files, it is recommended that the SWID tag data be stored in the API managed repository as well as stored as a file on the system. This allows older discovery tools that may not utilize API support to continue to be able to identify SWID tag files on a device.

Finally, the SWID tag data may also be accessible via a URI, or other means (such as using a common information model, e.g. Reference [12]).

The platform provider may provide access to the software identification tag using methods that are available through APIs, RPCs, command line interface, or other programmatic means.

6.1.5 Unique registration ID (regid)

6.1.5.1 Overview

Software identification tags may be created by different organizations and do not strictly require a centralized registration authority. Additionally, this part of ISO/IEC 19770 allows entities to create software identification tags for software components they did not create (such as an organization creating software identification tags for their internal software discovery processes). To accommodate these requirements, this part of ISO/IEC 19770 uses a regid. The regid provides a unique naming authority identifier.

6.1.5.2 Structure of regid

A regid shall use a URI reference (see IETF RFC 3986). Once an organization specifies a regid for their organization's software, that regid shall be used consistently for all software from the organization.

To ensure interoperability, allow for open source project support and third party tag consistency, the following recommendations should be applied when creating a regid.

— Unless otherwise required, the URI should utilize the http scheme.

— If the http scheme is used, the "http://" may be left off the regid string (a string without a URI scheme specified is defined to use the "http://" scheme).

— Unless otherwise required, the URI should use an absolute-URI that includes an authority part, such as a domain name.

— To ensure consistency, the absolute-URI should use the minimum string required (for example, example.com should be used instead of www.example.com).

6.1.5.3 Examples of regid

A regid for a company that creates and sells software is expected to be the HTTP reference to that company. So a regid for the Fabrikam Company is

"fabrikam.com"

A regid for an open source project called SampleProject that is hosted on soureceMyProject.net may be one of the following:

sampleproject.sourcemyproject.net

or