BS ISO/IEC 19770-2:2015
Incorporating corrigendum July 2018

BSI Standards Publication

Information technology — Software asset management

Part 2: Software identification tag

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 19770-2:2015

National foreword

This British Standard is the UK implementation of ISO/IEC 19770-2:2015. It supersedes BS ISO/IEC 19770-2:2009, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software and systems engineering.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2018
Published by BSI Standards Limited 2018

ISBN 978 0 539 01864 6

ICS 35.080

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2015.

Amendments/corrigenda issued since publication

Date	Text affected
31 July 2018	Implementation of ISO corrected text 15 February 2017: See ISO foreword for details

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 19770-2:2015

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)

Documents
Document Sharing Hub
© ISO/IEC 2015

Full document details: BS ISO/IEC 19770-2:2015

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 19770-2:2015

Contents

Page

Foreword ... **v**

Introduction ... **vi**

1 Scope ... **1**

2 Normative references ... **1**

3 Terms, definitions, and abbreviated terms ... **2**
3.1 Terms and definitions ... 2
3.2 Abbreviated terms ... 2

4 Conformance ... **3**
4.1 SWID tag conformance ... 3
4.2 Application conformance ... 3
4.3 Platform conformance ... 3

5 Interoperability guidance ... **3**
5.1 Overview ... 3
5.2 SWID tag modification ... 3
5.3 SWID tag relationships ... 4
5.3.1 Overview ... 4
5.3.2 Pre-installation data attribute ... 4
5.3.3 SWID patch attribute ... 4
5.3.4 SWID supplemental attribute ... 5

6 Implementation of software identification tagging processes ... **6**
6.1 General requirements and guidance ... 6
6.1.1 XML and XSD ... 6
6.1.2 SWID tags based on earlier revisions of this part of ISO/IEC 19770 ... 6
6.1.3 SWID tag installation and removal ... 6
6.1.4 SWID data storage and transmission ... 6
6.1.5 Unique registration ID (regid) ... 7
6.1.6 Tag identifier ... 8
6.1.7 Unique software identification tag file name ... 8
6.1.8 Software identification tag discovery ... 8
6.1.9 Languages ... 8
6.1.10 Authenticity of software identification tags ... 9
6.1.11 File hash definitions ... 9
6.1.12 Use of standardized data types in XSD definition ... 10
6.1.13 Using Evidence or Payload ... 10
6.1.14 Redistributable software components ... 10

7 Platform requirements and guidance ... **10**

8 Elements ... **11**
8.1 General ... 11
8.2 Minimum SWID tag data values required ... 12
8.3 Recommended SWID tag data values ... 13
8.4 XML element and attribute names ... 13
8.4.1 Introduction ... 13
8.4.2 Additional attributes allowed ... 14
8.5 Data values ... 14
8.5.1 SoftwareIdentity ... 14
8.5.2 Entity ... 18
8.5.3 Evidence ... 20
8.5.4 Link ... 20
8.5.5 Meta ... 25
8.5.6 Payload ... 26
8.6 Type and attribute definitions ... 26

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 19770-2:2015

8.6.1 Directory ... 26
8.6.2 File ... 27
8.6.3 FileSystemItem ... 28
8.6.4 Ownership ... 30
8.6.5 NMTOKEN and NMTOKENS ... 30
8.6.6 Process ... 30
8.6.7 Rel ... 30
8.6.8 Resource ... 31
8.6.9 ResourceCollection ... 31
8.6.10 Role ... 32
8.6.11 SoftwareMeta ... 32
8.6.12 Use ... 34
8.6.13 VersionScheme ... 35

Annex A (informative) **XSD changes between revisions** ... **36**

Annex B (normative) **XML schema definition (XSD)** ... **39**

Annex C (informative) **UML structure of SWID tag schema** ... **60**

Annex D (informative) **Sample tags** ... **62**

Bibliography ... **73**

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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*

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

Documents
Document Sharing Hub

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 2015 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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

© ISO/IEC 2015 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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.

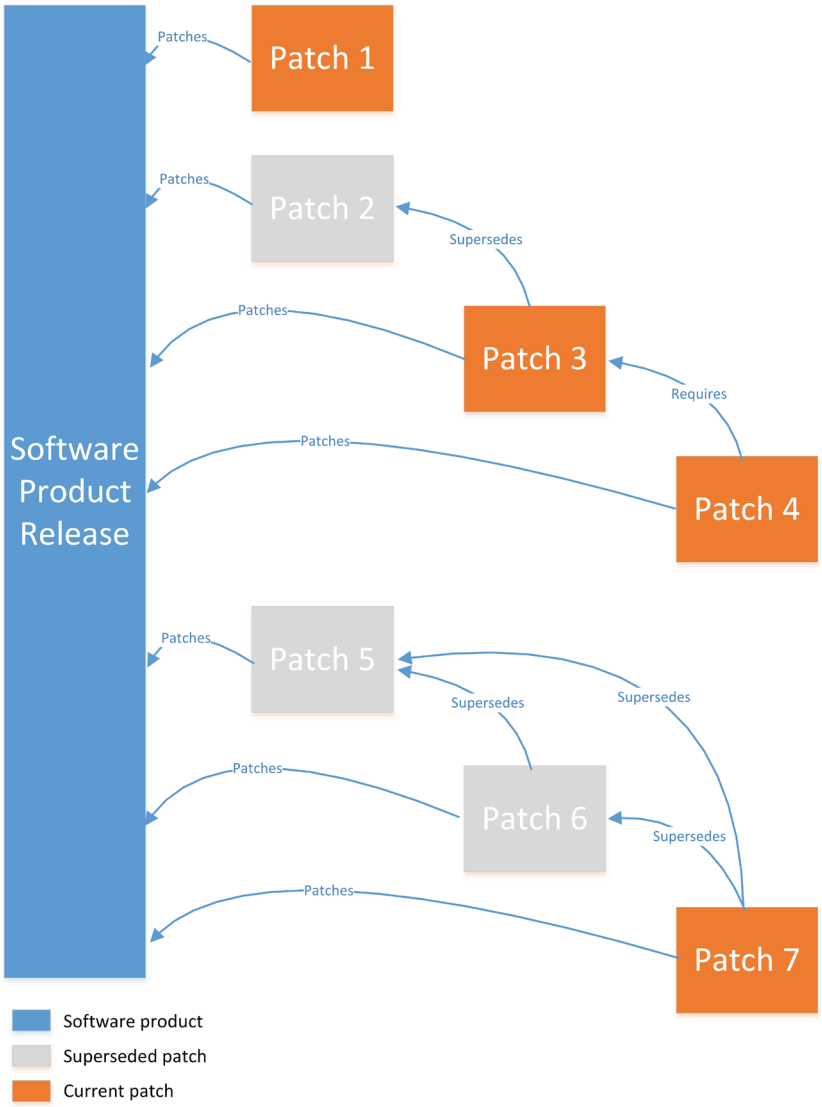

Figure 1 — Representation of patch and product relationships

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.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

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

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

sourcemyproject.net/sampleproject

or

sourcemyproejct.net/?projectname=sampleproject

The appropriate regid to choose for the project is dependent on the default reference used by the hosting site for the particular project.

6.1.6 Tag identifier

For the purposes of uniqueness, tagId shall be a globally unique value that shall be unique for every SWID tag created. It is recommended that tagId be a minimum of a 16 byte globally unique identifier (GUID), but it may also be a string that is constructed by the tag provider or another globally unique value. If a string construct is used, the tagId shall follow the restrictions for URI character use as specified in IETF RFC 3986, characters. For more information on universally unique identifiers (of which a GUID is one type of UUID), see Reference [2].

6.1.7 Unique software identification tag file name

SWID tag files may be located on the installation medium or on a device where software has been installed. There is no requirement to use different filenames that identify software in a pre-installation state as compared to tags located on a computing device due to software being installed or discovered. The difference between SWID tags for software in a pre-installation state and software that is installed is that software in a pre-installation state will have the attribute corpus set to true.

The following rules and recommendations shall be used when creating filenames.

Filenames should be restricted to use only the characters listed in the Portable Filename Character Set defined in IEEE 1003.1:2013, 3.278 to maximize interoperability between platforms. If this limitation is too restrictive, the tag creator shall ensure that the characters used in the filename are valid characters for all platforms where their SWID tags may be stored on a file system.

SWID tag base filenames (i.e. the filename without the .swidtag extension) shall be structured to be globally unique for the tag creator and product.

SWID tag creators may use different approaches to defining the base portion of the SWID tag filename; however, if the filename aligns with the following structure, the filename will be unique for the product and recognizable by a system administrator

<name of the tag creator> + <product name>.swidtag

The .swidtag file extension shall be used for all software identification tags.

6.1.8 Software identification tag discovery

The primary requirement is that the SWID tag shall be discoverable and be able to be read on the machine, distribution medium, and/or virtual environment on which the software package is installed by a suitably privileged user and/or discovery process. This includes the ability to utilize SWID tag information prior to software being installed to allow for capabilities such as validating installation files prior to software being installed on a system.

6.1.9 Languages

This part of ISO/IEC 19770 recognizes that software creators produce software with builds that are specific for a particular language while other software creators produce software with one build that implements add-on “language packs”. However, this part of ISO/IEC 19770 does not require software identification tags to recognize different language versions of the same product. Language information will be more efficiently managed through the use of a supplemental tag where the supplemental tag indicates specific language support and references the primary SWID tag.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

See Annex D for sample tags.

For encoding purposes, the use of UTF-8 (see Reference [15]) is the suggested methodology for software identification tags created based on this part of ISO/IEC 19770 (see Reference [14]).

6.1.10 Authenticity of software identification tags

Authenticity of an SWID tag, for example, to validate that the tag collected during a discovery process has not had specific elements of the tag altered, may be proven through the use of digital signatures within the tag.

Signatures are not a mandatory part of the software identification tag standard and can be used as required by any tag producer to ensure that sections of a tag are not modified and/or to provide authentication of the signer. If signatures are included in the software identification tag, they shall follow the W3C recommendation defining the XML signature syntax which provides message integrity authentication, as well as signer authentication services for data of any type.

The W3C Recommendation, XML Signature Syntax, and Processing Version 1.1 should be used with the canonicalization version 1.1 algorithm for digital signatures. Tool providers should be aware that Version 1.1 of the signing process is likely to have a signature corrupted if there are changes in whitespace within the signed elements of a SWID tag. This is primarily a consideration for discovery tools and systems. The recommendation is that at least one master, unaltered copy of each SWID tag be retained in the tools database for signature validation purposes.

When a signature is utilized for a SWID tag, the signature shall be an enveloped signature and the digital signature shall include a timestamp provided by a trusted timestamp server. This timestamp shall be provided using the XAdES-T form. Information on this form of timestamp can be found in Reference [13]. The SWID tag shall also include the public signature for the signing entity.

The requirement for a digitally signed SWID tag is that a SWID tag consumer shall be able to utilize the data encapsulated by the SWID tag to ensure that the digital signature was validated by a trusted certificate authority (CA), that the SWID tag was signed during the validity period for that signature and that no signed data in the SWID tag has been modified. All of these validations shall be able to be accomplished without requiring access to an external network. If a SWID tag consumer needs to validate that the digital certificate has not been revoked, then it is expected that there will be access to an external network or a data source that can provide revocation information.

For additional information on digital signatures, how they work and the minimum requirements for digital signatures used for US Federal Government processing, please review Reference [6].

6.1.11 File hash definitions

There are multiple different file hash algorithms in use throughout the industry. These algorithms have a variety of benefits and are used by a variety of different organizations. This part of ISO/IEC 19770 does not mandate that one version of a hash algorithm be utilized. The following are commonly used XML namespace references for typical hash algorithms currently in use:

— xmlns:MD5="http://www.w3.org/2001/04/xmldsig-more#md5";

— xmlns:SHA-1="http://www.w3.org/2000/09/xmldsig#sha1";

— xmlns:SHA-224="http://www.w3.org/2001/04/xmldsig-more#sha224";

— xmlns:SHA-256="http://www.w3.org/2001/04/xmlenc#sha256";

— xmlns:SHA-384="http://www.w3.org/2001/04/xmldsig-more#sha384";

— xmlns:SHA-512="http://www.w3.org/2001/04/xmlenc#sha512";

— xmlns:RIPEMD-160="http://www.w3.org/2001/04/xmlenc#ripemd160".

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

Additional hash algorithms may be developed and used over time and organizations may use the appropriate hash algorithm for their needs.

For security focused use, a minimum of the SHA-256 algorithm is recommended at the time of publication for this part of ISO/IEC 19770. The minimum algorithm for hashes may change as new algorithms are developed, validated, and programmatic libraries become available.

6.1.12 Use of standardized data types in XSD definition

There are a number of standardized types used in the software identification tag including specific date/time entries that shall follow formats as specified in the W3C recommendation titled, "*XML Schema Part 2: Datatypes Second Edition*". By using specific types as specified in the W3C recommendation, software identification tags can utilize an automated validation step that allows a much more consistent structure to the data provided.

6.1.13 Using Evidence or Payload

The evidence element (8.5.3) is similar to a payload element (8.5.6) in that these two elements contain information about resources (files, execution processes, and other system settings) that may be installed or discovered on a device. There is a key difference between Evidence and Payload.

The Payload element indicates the resources that may be installed on a device when the software is installed. Data in this subclause can be used, for example, to determine if files in a directory match the files that were designated as being installed with a software component or software product.

The Evidence element is used by discovery tools that identify software that does not include a SWID tag, and for which the tool creates a SWID tag based on discovery data. In this case, the element Evidence indicates what was discovered on the device, but it is not data that can be used to determine if files match what a publisher specified since there was no SWID tag to begin with. Evidence may be provided by systems that are discovering software dynamically and providing the data to a server, or may be used by a system to send data to a server as evidence of what was found. In the case where a SWID tag with the evidence element included is provided to the server, data from the evidence elements will be analysed either manually or in an automated fashion and a SWID tag identified as appropriate for the collected evidence. This SWID tag will be provided with the expectation that the server will provide that SWID tag back to the client to be placed on the device indicating that a specified software component or product was identified by the discovery tool.

6.1.14 Redistributable software components

Software creators regularly develop software components that may be redistributed by other organizations. Redistribution is done through entitlements and may include, but is not limited to, organizations making redistributable libraries and organizations making OEM software.

In either case, specifics of what is included in the redistributable package (for example, who is listed as the software creator and who is the tag creator) are agreed to by the parties involved and are not addressed in this part of ISO/IEC 19770.

Once agreed, SWID tags that are distributed through redistributable software components may be referenced just as any other SWID tag is referenced using the Link element. This allows, for example, a language library that is redistributed by an operating system manufacturer to be distributed by a tool provider with the tool including a "Link" element identifying a relationship to the library. See 8.5.4 for more information on the Link element and Annex D for examples of how redistributable packages may be referenced by a software product.

7 Platform requirements and guidance

The information contained within a software identification tag should be independent of the platform on which the software component that the tag references is installed. A platform, for example an

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

operating system, virtual environment, or application platform, should define processes to store and retrieve these software identification tags efficiently.

Software identification tags may be managed using different methods depending on the options a platform makes available for SWID tag management, as well as the options the software installation process utilizes for SWID tag management. The options shown in Table 1 may exist for every platform with the expectation that over time, more platforms and more software installation scripts will utilize the more effective options for SWID tag management and reporting:

Table 1 — Platform guidance for access to SWID tags

Method	Description
Programmatic Interface (SWID tag APIs)	A programmatic interface allows application software to call an operating system level service to manage a SWID tag from a particular platform. If a programmatic interface is supported and the platform supports the concept of a file system, a SWID tag should still be stored as a file in the software applications program file directory (see 6.1.4) to ensure that tag consumers who rely on access to the files can still process SWID tag data properly. The data provided in the SWID tag can additionally be stored in a data storage location as specified and developed by the platform owner. Using this approach, when an installation process uses the interface, additional information can be tracked and retained for all software installed, patched, upgraded, and removed from a system. Additional information that may be tracked includes the following: — user id used to install or remove a software component; — whether the installation process was initiated from the local system or remotely; — installation instance id (if there is more than one installation); — location of the installed software; — date and time software was installed or removed from the device. A programmatic interface can also provide a more effective, robust, and efficient software discovery process. Efficiencies may include the following: — compressing all transmitted SWID tag data; — transmitting SWID tag data only when the data changes; — automatically validating that all software on a device has an appropriate level of SWID tag as defined by tag consumer policy (for example, ensuring that all SWID tags are digitally signed); — automatically validating if executable files are stored on a device, but are not included in any SWID tag package_footprint.
SWID tag file stored with installed application	When a SWID tag file is stored in the file system of an application, the SWID tag file shall be located on the 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 the SWID tag represents. It is recommended, but not required, that the swidtag directory is located at the top of the application installation directory tree whenever possible. Any payload information provided must reference files using a relative path of the location where the SWID tag is stored. EXAMPLE If a program called Fabrikam Suite is installed on a device in the "c:\Program Files\Fabrikam Suite" directory, the SWID tag will be located in the following directory: C:\Program Files\Fabrikam Suite\swidtag

8 Elements

8.1 General

It is recommended that tag producers maintain a central repository of all software identification tags created for all product releases. This repository can then be used to validate the uniqueness of GUIDs (see ISO/IEC 19770-5, as well as the details for tagId in 8.5.1) and validate that other elements

© ISO/IEC 2015 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

are normalized such as the roles; softwareCreator, tagCreator, and licensor. This part of ISO/IEC 19770 does not require an external registration agency for software identification tags, so it is up to each tag producer to ensure each of their tags is unique.

Data values and types are defined in 8.5 and 8.6. The examples are specified in XML syntax with the format that shall be used for software identification tag creation. The examples provide additional insight as to what information is to be included within data values.

Data values (see 8.5) are XML elements of the SoftwareIdentity element that provide the specific identification information about a software component. Data values include details such as the name of the software component, the version, if the software component is a patch, etc.

Data types (see 8.6) are XML types utilized by the data elements and attributes.

8.2 Minimum SWID tag data values required

Due to the multiple use cases identified for SWID tag creation, the minimum data requirements for a SWID tag are relatively sparse. The only values that are required for a SWID tag to be considered "valid" to meet the requirements of the XML schema shall be the following:

— SoftwareIdentity (8.5.1):

  — name;

  — tagId;

— Entity (8.5.2):

  — role of TagCreator (at a minimum);

  — regid of TagCreator (at a minimum);

  — name of TagCreator (at a minimum).

In addition, there are a number of SWID tag attributes in the SoftwareIdentity element that have default values. These include the following:

— SoftwareIdentity (8.5.1):

  — Entity.regid – default value is http://invalid.unavailable;

  — patch – default value is false;

  — supplemental – default value is false;

  — tagVersion – default value is 0;

  — version – default value is 0.0;

  — versionScheme – default value is multipartnumeric.

These default values are specified so that if no value is included for these attributes, the SWID tag is considered to be the first version of a primary tag and that the software product has the version number of 0.0.

Although a SWID tag specifies only the software component name, the tagId and tagCreator is a valid SWID tag file that does not mean that the information is sufficient for processes where SWID tags are used. 8.3 provides details on which SWID tag data IT organizations will expect to see included in the SWID tags they receive.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

8.3 Recommended SWID tag data values

Due to the various use cases that apply to a SWID tag, there are a number of data attributes that are specified as optional because they may not be available in certain cases (for example, an organization may not specify a version number for a product – it is unusual, but possible that this element will be unspecified or unknown by a tool or service provider who may be creating a SWID tag for a software product).

For most software products, however, the data elements specified below should be made available whenever possible. This allows organizations to more completely automate their IT processes.

a) SoftwareIdentity (8.5.1):

   1) name;

   2) tagVersion;

   3) tagId;

   4) version;

   5) versionScheme;

b) Entity (8.5.2):

   1) name;

   2) regid;

   3) role (minimum requirement is that the tagCreator role is specified).

As well, the following data should be provided whenever possible to assist in IT automation processes:

a) SoftwareIdentity (8.5.1):

   1) patch (if this tag provides data about a patch);

   2) supplemental (if this tag provides supplemental information to another tag);

b) Entity (8.5.2 whenever possible, provide the softwareCreator and licensor role data as well);

c) Meta (8.5.5);

   1) product (this is the product name only, not including the edition, colloquial version, or services pack information).

If the software specified in this SWID uses the following type of naming structures, they should be provided both for human usability and so that other IT systems can utilize the information for automation purposes:

— edition;

— colloquialVersion;

— revision.

8.4 XML element and attribute names

8.4.1 Introduction

Software identification tag content shall be identified in accordance with the XML element and attribute names specified in 8.5. This naming requirement ensures consistent interoperability of software identification tag content, regardless of the creator or consumer of the tag data.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

Data values in the SWID tag may be structured as XML elements (which contain XML attributes), or XML attributes (which contain the actual data values). The following convention for the left-hand column in the tabular structures in 8.5 and 8.6 differentiates attributes and Elements.

Elements are Pascal-cased (ThisIsAnExample)

Attributes are camel-cased (anotherExampleIsLikeThis)

Additionally, required elements are shown in bold.

Unless specified otherwise, elements may be repeated as many times as required by the SWID tag creator.

8.4.2 Additional attributes allowed

This part of ISO/IEC 19770-2 allows the following W3C attributes to be used for any element:

— lang – this attribute specifies the language of the string specified in the element where it is specified as well as defining the lang for all child elements unless specified otherwise.

— id – this attribute provides a unique id for the element that is unique within the XML file. The id attribute may be used for XML XPATH arguments, or for other XML processing as required by tool vendors.

— <namespace:any> - any other attribute can be included for an element if the attribute is defined with a namespace. This allows vendors, customers and tools to include unique attributes that may or may not be generally known others in the market.

The above changes clarify the intent of the attributes that are available to use, includes the id attribute and indicate to all readers that any other attributes can be included if desired.

8.5 Data values

8.5.1 SoftwareIdentity

Element name	SoftwareIdentity		
Description	Represents the root element specifying data about a software component. NOTE A software product may be made up of one or multiple software components. Also, Software components may be atomic, or may be made up of multiple components. Each component will have its own SWID tag and only one SoftwareIdentity will exist for any one component.		
attribute or Element	**Type**	**Required/ default**	**Definition**
Entity	**Entity** (see 8.5.2)	**An entry for "tagCreator" is required, all others are optional**	**Element that specifies the organizations responsible for this SWID tag.**

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

Evidence	ResourceCollection NOTE Evidence references the section titled Evidence to differentiate it from Payload; however, it uses the type ResourceCollection, and adds two values – date and device. (see 8.5.3)	optional	Element used to provide results from a scan of a system where software that does not have a SWID tag is discovered. This type of information is not provided by the software creator and is instead created when a system is being scanned and the evidence for why software is believed to be installed on the device is provided in the Evidence element. NOTE Evidence is the same as payload with the exception that it also includes the date and device the evidence was collected.
Link	Link (see 8.5.4)	optional	Element to reference any other item (can include details that are related to the SWID tag such as details on where software downloads can be found, vulnerability database associations, use rights, etc.). NOTE This is modelled directly to match the HTML [LINK] element; it is critical for streamlining software discovery scenarios that these are kept consistent.
Meta	SoftwareMeta (see 8.6.11)	optional	Element that allows arbitrary key/value pair data related to this SWID. The attributes shown below are predefined attributes to ensure common usage across the industry. The schema allows for any additional attribute to be included in a SWID tag, though it is recommended that industry norms for new attributes are defined and followed to the degree possible.
Payload	ResourceCollection NOTE Payload references the section "Payload", to differentiate it from Evidence; however, it utilizes the type "ResourceCollection". Payload (see 8.5.6)	optional	Element that specifies the items that may be installed on a device when the software is installed. Note that Payload may be a superset of the items installed and, depending on optimization systems for a device, may or may not include every item that could be created or executed on a device when software is installed. In general, payload will be used to indicate the files that may be installed with a software product and will often be a superset of those files (i.e. if a particular optional component is not installed, the files associated with that component may be included in payload, but not installed on the device).
corpus	Boolean	false	If set to true, this attribute specifies that this SWID tag is a collection of information that describes the pre-installation data of software component.
patch	Boolean	false	If set to true, this attribute describes a product patch or modification to a different software element.
media	string (Media)	optional	'media' provides a way to describe the characteristics of the platform this SWID tag applies to (see the [Link] element 8.5.4 'media' attribute).

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015

name	**String**	**required**	This attribute provides the software component name as it would typically be referenced. For example, what would be seen in the add/remove dialog on a Windows device, or what is specified as the name of a packaged software product or a patch identifier name on a Linux device.
supplemental	Boolean	false	If set to true, this tag specifies supplemental tag data that can be merged with primary tag data to create a complete record of the software information. Supplemental tags will often be provided at install time and may be provided by different entities (such as the tag consumer, or a Value Added Reseller).
tagId	**String**	**required**	tagId shall be a globally unique identifier and should be assigned a GUID (see ISO/IEC 19770-5 definition for GUID). The tagId provides a unique reference for the specific product, version, edition, revision, etc. If two tagId's match and the tagCreator is the same, the underlying products they represent are expected to be the same. It is recommended that if a software release is available for multiple platforms, that each platform have its own unique tagId. However, it's recognized that some existing development processes may use the same tagId for the same release that may be available on different platforms. This allows IT systems to identify if a software item (for example, a patch) is installed simply by referencing the specific tagId value which is likely to be readily available in a software inventory. It is recommended, when possible, that a 16 byte GUID be used for this field; this provides global uniqueness without a significant amount of overhead for space. If use of a 16 byte GUID is not possible, a text-based globally unique ID may be constructed, this ID should include a unique naming authority for the tagCreator and sufficient additional details that the tagId is unique for the software product, version, edition, revision, etc. This would likely look as follows (+ is used as a string concatenation symbol): regid + productName + version + edition + revision + …

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 19770-2:2015