International Standard

ISO/IEC 19770-6

First edition
2024-01

Information technology — IT asset management —

Part 6: Hardware identification tag

Technologies de l'information — Gestion des actifs TI —
Partie 6: Étiquette d'identification du matériel

Reference number
ISO/IEC 19770-6:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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.

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

A list of all parts in the ISO/IEC 19770 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

0.1 Overview

The ISO/IEC 19770 series for information technology (IT) asset management (ITAM) addresses 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, the ISO/IEC 19770 series integrates tightly into all of the IT functions. Hardware identification (HWID) 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 the data interoperability of software and hardware management data, and the basis for many related benefits such as more effective security in the management of software and the authentication of hardware. ITAM standards for information structures also facilitate significant automation of IT functionality, such as improved authentication of software, and hardware for automated exposure of identification and mitigation.

0.2 Purpose of this document

This document is an International Standard for HWID tags. The hardware identification tag is a standardized data structure containing hardware identification information about a hardware product and/or the system configuration of multiple hardware products that supports new and automated management functions. Product information provided in the hardware identification tag structure is often provided in an XML data file, but the same HWID tag product information may be accessible through other means depending on the computing device being managed.

HWID tags are created by a HWID tag producer, for example, a hardware manufacturer who develops and distributes hardware. HWID tag data is utilized by HWID tag consumers, for example, an inventory tool or service that collects information from a physical or virtual device for a variety of purposes.

This document has been developed to facilitate automation of IT processes through the use of hardware identification tags and for applications which use those tags, for the purposes of inventory control, configuration management, hardware security, or logistics. This document includes information which facilitates human understanding (such as model and colloquial version name), but it is unrealistic to expect to create, manage, and use hardware identification tags without the use of automated capabilities built into specialist or generalized tools. The extent to which such capabilities are provided by specialist commercial products, open-source-type products, or platforms themselves, depends on market developments over time.

This document supports IT asset management processes as defined in ISO/IEC 19770-1. This document is also designed to work together with other parts in the ISO/IEC 19770 series, including ISO/IEC 19770-2, ISO/IEC 19770-3, ISO/IEC 19770-4, and ISO/IEC 19770-5, which are International Standards for software identification, entitlement, resource utilization measurement, and overview/vocabulary.

This document provides a common set of terms and associated transport format to facilitate the management of IT hardware. The intended benefits include easier demonstration of proof of ownership, improved asset management, and improved security.

Furthermore, an additional benefit of having a standard for describing hardware components is to encourage the normalization by industries of names for, and the details of, different types of hardware. A common lexicon is critical to standardization and shared understanding of terminology. The terms in this document should form a part of that lexicon over time.

Hardware identification tags can benefit all stakeholders involved in the development, manufacturing, distribution, deployment, installation, and on-going management of hardware. Key benefits associated with hardware identification tags include the following.

a) The ability to consistently and authoritatively identify hardware products that need to be managed for any purposes of inventory control, configuration management, hardware security, or logistics or for the specification of dependencies. Hardware identification tags provide the meta-data necessary to support more accurate identification than other traditional hardware identification techniques.

b) The ability to identify groups of hardware products in the same way as individual hardware products (e.g., components and modules within a single system), thus enabling entire groups of hardware products to be managed as a system with the same flexibility as individual products.

c) The ability to automatically relate installed hardware with other information such as repair installations, configuration issues, maintenance agreements or vulnerabilities.

d) The ability to facilitate interoperability of hardware identification between different hardware manufacturers, different hardware platforms, different IT management tools, and within hardware manufacturing, as well as between HWID tag producers and HWID tag consumers.

e) The ability to facilitate automated approaches to hardware inventory, using information both from the hardware identification tag and from the software identification schema as specified in ISO/IEC 19770-2.

f) The ability to provide a comprehensive information structure that identifies different entities, including hardware manufacturers, packagers, distributors external to the hardware consumer, as well as various entities within the hardware consumer, associated with the system configuration, installation, and management of the product on an on-going basis.

g) The ability to establish trust through the optional use of digital signatures by organizations creating hardware identification tags, the ability to validate that hardware identification is authoritative, and from a trusted source.

h) The opportunity for entities other than original hardware manufacturers (e.g. independent providers or in-house personnel) to create non-authoritative hardware identification tags for legacy hardware, and/or for hardware from other manufacturers who do not provide hardware identification tags themselves.

Annex A contains the XML schema document for HWID tags; Annex B provides a UML diagram of the HWID tag schema; Annex C provides sample HWID tags.

Information technology — IT asset management —

Part 6:
Hardware identification tag

1 Scope

This document provides specifications for a transport format which enables the digital encapsulation of this data. This document refers to an encapsulation of hardware identification (HWID) data as a HWID tag, just as ISO/IEC 19770-2 refers to software identification (SWID) tags for software identification.

This document applies to the following.

— Tag producers: organizations that create HWID tags for use by others in the market. A tag producer can be part of the organization creating the hardware or a third-party organization. These organizations can be broken down into two major categories.

  — Device or component providers: entities responsible for the manufacturing or creation of the hardware device and/or associated operating system, virtual environment, or application platform. Platform providers which support this document can additionally provide tag management capabilities at the level of the platform or operating system.

  — Tag tool providers: entities that provide tools to create hardware identification tags. For example, tools within development environments that generate hardware identification tags, or installation tools that can create tags on behalf of the installation process, and/or desktop management tools that can create tags for underlying hardware, virtual machines, or platforms that did not originally have a hardware identification tag.

— Tag consumers: tools and/or organizations who utilize information from HWID tags are broken down into the following two major categories.

  — Device or component consumers: entities that purchase, install, integrate, and/or otherwise deploy physical or virtual hardware or components.

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

This document deals only with hardware device or component identification.

This document does not detail information technology asset management (ITAM) processes required for discovery and management of hardware (which is provided in ISO/IEC 19770-1) software identification tags (as defined by ISO/IEC 19770-2), entitlement tags (as defined by ISO/IEC 19770-3), or resource utilization measurements (as defined by ISO/IEC 19770-4).

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19770-2, *Information technology — IT asset management — Part 2: Software identification tag*

ISO/IEC 19770-3, *Information technology — IT asset management — Part 3: Entitlement schema*

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

RFC 3986[1)], *Uniform Resource Identifier (URI): Generic Syntax*

RFC 7515[2)], *JSON Web Signature*

XML Signature Syntax and Processing Version 1.1, W3C Recommendation 11 April 2013 https://www.w3.org/TR/xmldsig-core1/

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.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1.1
dynamic attribute
element of a HWID tag that may change over the life of the product or are defined after creation

3.1.2
HAM
hardware asset management
coordinated activity of an organization to realize value from hardware assets

Note 1 to entry: Hardware asset management is a specialization and sub discipline of IT asset.

3.1.3
HWID creator
entity that initially creates a HWID record

Note 1 to entry: This entity can be part of the organization that manufactured the component to which the record relates, in which case the HWID creator and component manufacturer are the same. The HWID creator can also be a separate organization or third party unrelated to the manufacturer (such as in the case where HWID records are created for existing hardware components by an operating system or a tool deployed by the device owner).

3.1.4
HWID schema
hardware Identification schema
information structure containing a digital description of a hardware component and its associated information

3.1.5
legacy hardware
hardware originally created without native information structures

1) https://www.ietf.org/rfc/rfc3986.txt.

2) https://tools.ietf.org/html/rfc7515.

3.1.6
software identification tag
SWID tag
information structure containing identification information about a software configuration item.

[SOURCE: ISO/IEC 19770-5:2015, 3.40, modified — At the end of the definition, "which may be authoritative if provided by a software creator" has been removed.]

3.1.7
static attribute
element of a HWID that do not change over the life of the product or are defined at creation

3.2 Abbreviated terms

Ent	software Entitlement schema, Entitlement schema
GUID	globally unique identifier
HAM	hardware asset management
HWID	hardware identification
IETF	internet engineering task force
ITAM	information technology asset management
JSON	javascript object notation
OEM	original equipment manufacturer
regid	registration identifier
SAM	software asset management
SKU	stock keeping unit
SWID	software identification
UNSPSC	united nations standard products and services code
URI	uniform resource identifier
URL	uniform resource locator
W3C	World Wide Web Consortium
XML	extensible markup language
XSD	XML schema document

4 Conformance

4.1 HWID tag conformance

A hardware identification tag is in conformance with this document if the tag data structure meets all the requirements specified in this document.

4.2 Application conformance

Application conformance incorporates both syntax and semantics.

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

— A conforming tag producer shall be able to produce HWID tags conforming to this document.

— A conforming tag consumer shall treat the information in HWID tag in a manner consistent with the semantic definitions given in this document. An application's intended behaviour shall not require that application to process all of the information in a HWID tag. However, the information that it does process shall be processed in a manner that is consistent with the semantic definitions given in this document.

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

4.3 Platform conformance

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

5 Interoperability

5.1 Overview and key design decisions

This clause explains the essential nature of HWIDs, and how the different types of HWID records interrelate and are designed to provide for interoperability of creation and usage by all parties involved with hardware component data.

5.2 Hardware identifiers - <hwidID>

The unique identifier for each HWID is the `<hwidID>`. This is a GUID which may be formed in different ways, as long as global uniqueness is achieved. A 16-byte GUID shall be used for this field – this provides global uniqueness without a significant amount of overhead for space. These GUIDs should be generated in a fashion compliant to ISO/IEC 9834-8.

For hwidID, 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 `<entCreator>` and sufficient additional details that the `<hwidId>` is unique for the entitlement. This can look as follows (+ is used as a string concatenation symbol):

regid + productName + version + edition + revision + ...

5.3 Use case overview

There are a number of basic use cases which can be supported by the definitions contained in this document. Note that the HWID merely records the state of an artifact at a certain point in time. For example, a HWID represents a specific piece of hardware, in a specific location, with a specific owner at a specific point in time. To transfer hardware to another organization, it is not sufficient just to create transfer transactions recording a change in ownership and location using HWIDs; rather, compliance with the necessary terms and conditions (e.g. of the hardware maintenance) for the transfer is required.

5.4 HWID type

5.4.1 General

The main use cases supported by the HWID specification are provided in Table 1. Each provides a short explanation of how it is implemented.

The type of HWID is defined by the value hwidType.

Table 1 — hwidType values

hwidType value	Meaning
Primary	This is the “base” HWID tag for an asset, representing the primary attributes of the hardware being identified.
System	A tag establishing relationships between one or more primary and/or system tags for a given system. This is useful for connecting peripherals to a PC, virtual machines to a physical host or multiple systems together in a cluster.
Supplemental	Supplemental tags are designed to provide additional information to either primary or system tags.

5.4.2 Issuance of a primary HWID

Issuance of a primary HWID is the 'base' HWID type. It is expected that primary HWIDs should be issued by the hardware manufacturer, who should be able to provide authoritative information relating to the hardware’s details. HWIDs may also be issued by third party suppliers but would be considered a non-authoritative tag. This may be necessary if manufacturers do not supply HWIDs, such as to encapsulate legacy hardware which had no associated HWIDs, etc.

Primary HWID tags are implemented by creating a HWID of `<hwidType> = 'Primary'`.

5.4.3 Adding information to a primary HWID

End-user organizations and third parties may wish to add information to a HWID, e.g. order information, entitled entities and or user configured identifiers.

Information is added to a HWID tag by creating a HWID of `<hwidType> = 'Supplemental'`, and with `<supplementalHWIDType> = 'InfoAdded'` which specifies a <linkedToPrimaryhwidID> equal to the `<hwidID>` of the primary HWID which is to be extended.

Example of a primary HWID:

```
fabrikam.com_MightyPC_AAA.HWID
<HWID
    HWIDId=”AAA”
    hwidType=”Primary”
```

Example of a linked supplemental HWID:

```
reseller.com_MightyPC_BBB.HWID
<HWID
    HWIDID=”BBB”
    hwidType=”Supplemental”
    supplementalhwType=”InfoAdded”
<Link>
    linkedToPrimaryhwidID=”AAA”
</Link>
```

5.4.4 Archiving a primary HWID

The purpose of archiving is to remove HWIDs from active use, so that they are no longer valid for hardware management purposes. This is appropriate, for example, for hardware that is no longer being used by the organization and which is no longer required to be tracked as an asset.

The archive functionality is implemented by creating a supplemental HWID of `type = 'Archived'`, this identifies the `<hwidID>` of the HWID which is to be revoked in `<linkedToPrimaryhwidID>` attribute.

5.4.5 Issuance of a system HWID

Issuance of a system HWID combines multiple primary HWIDs into a logical grouping of an entity that is treated as a single instance. It is expected that system HWIDs should be issued by the hardware manufacturer, who should be able to provide authoritative information relating to the hardware details during the manufacturing or assembly process. System HWIDs may also be issued by third party suppliers or system integrators as non-authoritative tags. The issuance of non-authoritative system HWIDs may be necessary if manufacturers do not supply system HWIDs.

System HWID tags are implemented by creating a HWID of `<hwidType> = 'System'`.

5.4.6 Adding information to a system HWID

End-user organizations and third parties may wish to add information to a system HWID, e.g. addition of additional information about the system.

Information is added to a HWID tag by creating a HWID of `<hwidType> = 'Supplemental'`, and with `<supplementalHWIDType> = 'InfoAdded'` which specifies a `<linkedToSystemhwidID>` equal to the `<hwidID>` of the primary HWID which is to be extended.

Example of a system HWID:

```
fabrikam.com_MightyPC_BBB.HWID
<HWID
    hwidID=”BBB”
    hwidType=”System”
   <Link>
      linkedToPrimaryhwidID=”AAA”
   </Link>
```

Example of a linked supplemental HWID:

```
reseller.com_MightyPC_CCC.HWID
<HWID
    hwidID=”CCC”
    hwidType=”Supplemental”
    supplementalhwType=”InfoAdded”
   <Link>
       linkedToSystemhwidID=”BBB”
```

5.4.7 Archiving a system HWID

The purpose of archiving is to accomplish one of the following two goals.

— Remove system HWIDs from active use, so that they are no longer actively used for hardware management purposes. This would be appropriate, for example, for hardware that is no longer being used by the organization and which is no longer required to be tracked as a system.

— Remove system HWIDs where the makeup of the system has been altered sufficiently enough where it makes more sense to reissue a new system tag. At this point a new system tag may be generated.

The archive functionality is implemented by creating a supplemental HWID of `type = 'Archived'`, this identifies the `<hwidID>` of the HWID which is to be revoked in `<linkedToSystemhwidID>` attribute.

5.4.8 Systems of systems

As system tags define groups of hardware assets that logically belong together, system tags may also be used to link together logically grouped systems. For example, a data center server can be represented by a system HWID tag containing the server computer, the storage medium it uses, and a power supply. It may also be beneficial to think of the entire rack as a single unit of systems that includes multiple data center servers, a top of rack network switch, the rack itself, and all cables used to interconnect the systems.

A system of systems is described by creating a new HWID tag of `hwidType="System"`, and then using the `linkedToSystemhwidID` attribute to connect multiple system HWIDs.

Example of a primary system HWID:

```
reseller.com_MightyPC_ZZ.HWID
<HWID
    HWIDID="ZZ"
    hwidType="System"
   <Link>
    linkedToSystemhwidID="XX"
   </Link>
   <Link>
    linkedToSystemhwidID="YY"
   </Link>
```

Example of a subsystem HWID 1:

```
fabrikam.com_MightyPC_XX.HWID
<HWID
    HWIDId="XX"
    hwidType="System"
    <Link>
      linkedToPrimaryhwidID="AAA"
    </Link>
```

Example of a subsystem HWID 2:

```
fabrikam.com_MightyPC_YY.HWID
<HWID
    HWIDId="YY"
    hwidType="System"
    <Link>
    linkedToPrimaryhwidID="CCC"
    </Link>
    <Link>
```

5.5 Supplemental HWID types

If a HWID has a type of "Supplemental", their specific use is defined by the value `'supplementalHWIDType'`. Suggested values are given in Table 2. As per the definition of "`SupplementalHWIDType`" in 7.8.8, it is possible to define additional values.

This addition of additional values into supplemental HWID tags should only be done when specific use cases absolutely require the additional values to be contained within the HWID tag structure, as there is no common understanding of the definition for these additional values.

Table 2 — supplementalHWIDType values

supplementalHWIDType value	Meaning
InfoAdded	This supplemental record is to add information to an existing primary HWID. If different data values are added for the same element or attribute (whether in the same or in different supplemental HWIDs), then the interpretation of that data, i.e. the order of precedence, is not specified by this document. (Supplemental HWIDs with incorrect or changed values should be revoked – see below.)
Archived	This supplemental HWID archives an existing HWID. The <hwidID> for the HWID which is archived is specified in the <linkedToPrimaryhwidID> or <linkedToSystemhwidID> attribute. If the <linkedToPrimaryhwidID> or <linkedToSystemhwidID> value indicates a HWID which has the <hwidType> of "Supplemental" then only that supplemental HWID is archived. If the <hwidID> indicated in the <linkedToPrimaryhwidID> or <linkedToSystemhwidID> attribute is not of <hwidType> "Supplemental" then that <hwidID> and all supplemental HWIDs associated with that <hwidID> are to be considered archived. The primary reasons that a HWID would be marked archived are: — The HWID referenced needs to be replaced because of incorrect or missing information. Instead of adding the information with an InfoAdded, the entire HWID is archived and reissued. — The hardware or system described by the HWID referenced has been removed from inventory, and therefore no longer to be actively used for hardware management purposes. This would be appropriate, for example, for hardware resold, destroyed or is otherwise permanently unavailable. Note that hardware that is transferred to another entity might be better served by updating the entity information of the new owner to maintain history and continuity of the HWID tag. The effect of the 'Archived' supplemental HWID is the same as that of 'Revocation', but with an identified purpose.
<any>	Any other value of may be used, to provide extensibility in this document for additional uses not yet specified.

5.6 Key design decisions of HWIDs

There are some key design decisions which underlie the specifications of this document:

Non-modifiable HWID records. Once a HWID tag has been created and placed into use, existing records of HWIDs shall not be modified in any way, including via the addition of data. There are some situations which can be considered modifications, which are handled in the following ways, without violating this design principle that HWIDs shall not be modified:

A primary HWID is issued which is incomplete or contains an error. There are two ways of dealing with this situation.

— Full replacement. The incomplete or erroneous HWID shall be revoked by issuing a supplemental HWID which archives the HWID and a new HWID is created which replaces the HWID which was incomplete or in error. (See 5.4.4.)

— Amending by adding data. Another approach with this document is to issue a supplemental HWID with just the updated data. The interpretation in such a situation is to use the latest version of specific data – the newest HWID replaces previous versions which contain the same data. However, a HWID creator may specify an alternative interpretation, e.g. the supplemental HWID can be purely additive, containing net new information. (See 5.4.3.)

A system HWID is issued which is incomplete or contains an error. There are two ways of dealing with this situation:

— Full replacement. The incomplete or erroneous HWID shall be revoked by issuing a supplemental HWID which archives the HWID and a new HWID is created which replaces the HWID which is incomplete or in error. (See 5.4.7.)

— Amending by adding data. Another approach with this document is to issue a supplemental HWID with just the updated data. The interpretation in such a situation is to use the latest version of specific data – the newest HWID replaces previous versions which contain the same data. However, a HWID creator may specify an alternative interpretation, e.g. the supplemental HWID can be purely additive, containing net new information. (See 5.4.6.)

A supplemental HWID is issued which is incomplete, contains an error, or has substantially changed. This approach is appropriate for allocations recording ownership information but can apply to other types of Supplemental HWIDs. The way of dealing with this situation in this document is to revoke just the supplemental HWID, and then issue a corrected supplemental HWID (as needed). (See 5.5.)

In order to reduce complexity of audit trail recording, as with ISO/IEC 19770-3, the approach to be taken for this document is to mandate the non-modifiability of HWIDs, with any additions or changes needing to be made via supplemental HWIDs, each of which provides its own clear audit trail. Furthermore, all HWIDs which are supplied by one organization (e.g. a hardware manufacturer) for use by another (e.g. an end-user organization) should be signed in their entirety, which in any case prevents data from being added to them. (See 6.5.)

Due to the transactional nature of HWIDs, HWIDs should represent individual actions (within the organization) or transactions (between organizations) related to hardware and their associated components. HWIDs are not intended to show the current state of something which is potentially changeable. Current state can only be determined by forming a consolidated view of all relevant HWIDs, which is the task of ITAM tools, not of the HWID records themselves.

5.7 Uniqueness of identifiers

5.7.1 General

As with ISO/IEC 19770-2, this document does not require a registration authority, and therefore it relies on the use of GUIDs to ensure uniqueness of identifiers within relevant contexts.

5.7.2 Entity registration identification – regid

5.7.2.1 General

The most important globally unique identifier for ISO/IEC 19770-2 and ISO/IEC 19770-3 as also for this document is the Entity registration identifier, or 'regid'. The regid provides a unique naming authority identifier.

5.7.2.2 Structure of regid

A regid shall use a URI reference in accordance with RFC 3986. Once an organization specifies a regid for their organization, that regid shall be used consistently for all IT asset information structures created by that organization. To ensure interoperability, allow for open-source project support and 3rd party tag consistency, the following recommendations apply 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 (e.g. example.com should be used instead of www.example.com).

5.7.2.3 Examples of regid

A regid for a company that creates and sells hardware is expected to be the HTTP reference to that company. So, a regid for the fictional company "Fabrikam" can be:

"fabrikam.com"

5.7.3 Hardware identification tag ID (hwidId)

A 16-byte GUID shall be used for this field – this provides global uniqueness without a significant amount of overhead for space. These GUIDs should be generated in a fashion compliant to ISO/IEC 9834-8.

For hwidId, 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 `<hwidCreator>` and sufficient additional details that the `<hwidId>` is unique for the entitlement. This would likely look as follows (+ is used as a string concatenation symbol):

regid + productName + serialNumber + …

5.8 Interoperability Design Considerations

To coordinate multiple tags which may be referring to the same object, the use of identifying elements describing the object shall be leveraged by each tag instance. By leveraging key, preferably unique, characteristics of the target which this tag describes, it allows other tags of different types to refer to the same end point.

The primary way in which these key identification attributes for the HWID specification is via elements in the hwidMeta structure:

```
fabrikam.com_MightyPC_XX.HWID
<HWID
    HWIDId="XX"
    hwidType="System"
    <hwidMeta>
          hwType="Desktop Computer"
          modelName="MightyPC"
          modelNumber="10"
          serialNumber="9ace45b"
    </hwidMeta>
```

Elements used for object identification are likely attributes that do not change (or are incredibly infrequently changed) and would be known by the creator of the product at time on manufacture or installation:

— CPULongDescription

— CPU Count

— CPUManufacturer

— assetName

— assetID

— UUID

— Serial number

— unspscCodemodelNumber

6 Implementation of HWID processes

6.1 General

This clause provides guidance about the processes related to the creation and use of HWIDs, and how these may impact the content of HWIDs.

The information contained within a hardware identification tag should be independent of the platform on which the software component that the tag references is installed. A platform, for example, an operating system, virtual environment, or application platform, should define processes to store and retrieve these hardware identification tags efficiently. It should be noted that a HWID tag should continue to exist or be able to be generated even if the software running on the platform is deleted, formatted, and otherwise cleared. Refer to 6.7 for further guidance on local storage considerations.

Hardware identification tags may be managed using different methods depending on the options a platform makes available for HWID tag management, as well as the options in off platform management and storage.

6.2 Platform requirements and guidance

6.2.1 HWID tag implementation considerations

Information that can be potentially stored in a HWID tag may be divided into two categories:

— Static attributes:

  Elements that are static typically would not change over the life of the product, however any change would typically mean a different HWID entry. Static attribute values are intrinsic to the uniqueness of a HWID, and any modification of static attributes shall be resolved by the reissuance of the HWID tag.

  Examples of static attributes:

  — serial number
  — GUID
  — product type
  — description
  — model name
  — model number
  — SKU
  — HWID creator

— Dynamic attributes:

  Elements that are dynamic typically would change over the life of the product or are defined after creation. The platform would have no way of knowing this information at creation time and would have to be provided as a supplemental entry later.

  Examples of dynamics attributes:

  — order info
  — device status
  — location
  — Channel type

— Entitled entity

— User configured identifiers (e.g. hostname)

— dates

— system tags

— links

6.2.2 HWID tag methods

The options shown in Table 1 may exist for every platform with the expectation that over time, more platforms and more platform enablement scripts will utilize the more effective options for HWID tag management and reporting (see Table 3).

Table 3 — Platform guidance for access to HWID tags

Method	Description
Programmatic interface (Primary HWID)	A programmatic interface allows application software to call an operating system level service to manage a HWID tag from a particular platform. A HWID tag using only static attributes should be able to be provided even if a local storage system has been erased or does not exist, however it is important that if this is the case, assuming no hardware changes, the API provides exactly the same HWID tag once invoked. This would be used by primary HWID types but can potentially be used by system HWID or supplemental HWID tags as well. A programmatic interface to retrieve a primary HWID is highly recommended for all implementation types as well as an interface to create a new primary HWID based off static attributes should the primary HWID no longer be available.
Programmatic interface (Supplemental and system HWID)	If a programmatic interface is supported and the platform supports the concept of a file system, a HWID tag should still be stored as a file in the software applications program file directory, to ensure that tag consumers who rely on access to the files can still process HWID tag data properly. The data provided in the HWID 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 in supplemental and system tags with additional information that may not always be available or is learned after the manufacturing and assembly process. Additional information that may be tracked includes the following: — User configured information such as — hostnames — interface addresses — configurations — Commerce information such as — ownership entities — partner information — sales order and purchase order information — Location information such as — site ids — physical address — responsible parties — Date and time of lifecycle events
HWID tag file stored with operating system	When a HWID tag file is stored in the file system of an application, the HWID tag file shall be located on the device's file system in a sub-directory named "hwidtag" (all lower case) that is located in the same file directory or sub-directory of the install location of the operating system of the platform that the HWID tag represents. The hwidtag directory should be located at the top of the OS installation directory tree whenever possible. Any payload information provided shall reference files using a relative path of the location where the HWID tag is stored.

6.3 HWID creators

Anyone, or any organization, may create HWIDs. The strong preference is for primary HWIDs to be created by the hardware manufacturer, so that these HWIDs have the highest degree of trustworthiness vis-à-vis the information they contain. However, there can be no assurance that all manufacturers will produce (authoritative) HWIDs, firstly for new hardware, and secondly for hardware already deployed to end user organizations. Therefore, it shall be possible for end-user organizations and third parties to create (non-authoritative) HWIDs themselves. Furthermore, there are certain types of management transactions which

would normally only be created by end-user organizations, but likewise these can also be produced by third parties depending on the circumstances.

The entity which creates a HWID is defined in the “entity” element, where `<role> = 'hwidCreator'`.

6.4 Trustworthiness of HWIDs

HWIDs can never be assumed to have complete trustworthiness. Primary reliance shall always be placed on physical examination of the hardware in question. Given this caveat, the trustworthiness of HWIDs is dependent on two components:

— Authority. Trustworthiness depends on the authority of the person or organization creating the HWID, for the information given in that HWID. For example, the hardware manufacturer would be expected to have the highest level of authority for creating a HWID for a piece of hardware it has sold, and therefore this type of information would have the highest degree of trustworthiness.

— Authentication. The information in a HWID needs to be authenticated to be certain of the level of trustworthiness which would be expected for the HWID creator. The expectation is that HWIDs are signed to provide such authentication, at a minimum for HWIDs which are created by one organization for use by another organization.

The attribute `<trustLevel>` allows a HWID creator to accord a level of trustworthiness (in their opinion) for a HWID. Refer to 7.8.9 for more information.

6.5 Authenticity of HWIDs

6.5.1 General

Authenticity of a HWID shall be able to be determined through the use of digital signatures.

Signatures are not mandatory for this document but are highly recommended and can be used as required by any HWID creator to ensure that sections of a HWID are not modified and/or to provide the authentication of the tag signer.

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

For additional information on digital signatures, refer to RFC 3986, RFC 7515, XML Signature Syntax and Processing Version 1.1 and References [5] to [14].

6.5.2 XML digital signatures

If signatures are included in the HWID, 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.

XML Signature Syntax and Processing Version 1.1 shall 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 HWID. At least one master, unaltered copy of each HWID should be retained in the tools database for signature validation purposes.

When a signature is utilized for a HWID, 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 the W3C note Entitled XML Advanced Electronic Signatures.

The HWID shall also include the public signature for the signing entity.

6.5.3 JSON digital signatures

If signatures are included in the HWID, they shall follow RFC 7515 defining JSON signature syntax which provides message integrity authentication as well as signer authentication services for data of any type.

6.6 HWID file names

If a HWID is transported or stored in a file and it is the only HWID in the file, the name of this file shall be created according to the following pattern: `<hwidCreatorRegid>_<product>.HWIDtag`.

The HWIDtag file extension shall be used for all hardware identification tags.

If multiple HWID records are transported or stored in a file, the name of this file shall be created according to the following pattern: `<hwidCreatorRegid>_multi.HWID`.

NOTE `<hwidCreatorRegid>` above is replaced with the value for regid for the hwidCreator entity.

6.7 HWID storage

It is expected that HWIDtag files are stored in a central location per device. This collection of HWIDtag files are referred to as a HWID Library. The specific location of the HWID files may vary, however the directory/ folder in which HWIDtag files are kept shall always be named "hwidtag". This facilitates rapid identification of HWID libraries on a device or external filestore.

Where devices do not contain files, it is expected that they offer the HWID information via an alternative method, such as an API or web interface, which should be offered via secure (e.g. HTTPS) based communication methods.

More likely than not, HWID tags beyond the primary tags are stored by an IT asset management solution. See 6.9 for more details.

6.8 HWID recovery

The HWID library should be able to function as a complete audit trail of hardware changes over time related to the device in question. HWID creators may or may not be able to provide backup copies of HWIDs (with identical <hwidID>'s and HWID data) to end-users. (See 7.2 for HWID creator recommendations.) In the event that the local data storage is deleted, it is the responsibility of the end-user to back up this data appropriately, potentially by using an off platform HWID management system or another appropriate system.

6.9 HWID Considerations for asset management platforms

The best practice for the HWID primary tag includes it focusing on the data elements that are immutable and extremely unlikely to be modified over the lifespan of the hardware. While dynamic attributes can also be stored in a local tag structure, they likely require multiple changes over the lifespan of the device and as such should be managed in an asset management platform instead of on the device directly.

For example, this includes items like: static attributes listed in 6.2.1, as well as examples in Clause C.1.

7 Hardware ID file data specification

7.1 General

This document does not require a specific process for generating content for HWID files.

HWID creators should maintain a central repository of all HWIDs created. This repository can then be used to validate the uniqueness of GUIDs (see ISO/IEC 19770-5 as well as the details for <hwidID> in 5.2) and validate that other elements are normalized such as the <role> values – hardwareManufacturer, hwidCreator, etc. This document does not require an external registration agency for HWID schema documents, so it is up to each HWID creator to ensure each of their HWIDs is unique.

Data definitions, enumerated values, and simple types are defined in 7.7 and 7.8. The examples are specified in XML and JSON syntax, with the format that shall be used for HWID schema document creation. The examples provided give additional insight as to what information is to be included within HWIDs.

Data definitions (7.7) specify the elements and attributes of the HWIDs that provide the information about a piece of hardware.

Attribute value definitions (7.8) give the specifications for types which have mandatory or recommended values.

This document (for the HWID schema) is intended to align closely with ISO/IEC 19770-2 (software identification tags) and ISO/IEC 19770-3 (entitlement schema). Any of the elements, attributes, or other specifications of ISO/IEC 19770-2 or ISO/IEC 19770-3 which the HWID creator may wish to utilize may be used in the hardware schema as well. Any use of SWID or Ent elements or attributes in the HWID shall conform with the relevant requirements from ISO/IEC 19770-2 and ISO/IEC 19770-3 respectively, unless otherwise specified in this document.

7.2 Minimum HWID tag data required

Due to the multiple use cases identified for HWID creation, the minimum data requirements for a HWID are relatively sparse and are identified by a 'requirement level' of 'M1' in the data definitions (7.7). The only values that are required for a HWID to be considered “valid” to meet the requirements of the XML or JSON schema shall be:

HWID (7.7.2)

— <hwidID>

— <HWIDCreationDate>

— <hwidType>

Entity (7.7.4)

— <role> of hwidCreator (at a minimum)

— <regid> of hwidCreator (at a minimum)

— <name> of hwidCreator (at a minimum)

Although a HWID that specifies only the hardware Type, the <hwidID> and hwidCreator is a valid HWID file, that does not mean that the information is sufficient for processes where HWIDs are used. Enough information shall be provided so that the HWID uniquely and accurately describes a piece of hardware. Information shall also be provided for the HWID to have value to a consumer (see 6.2.1). 7.7.3 (HWIDMeta) provides details on which HWID data IT organizations expect to see included in the HWIDs they receive.

Due to the various use cases that apply to a HWID, there are a number of data elements and attributes that are specified as mandatory only in specific cases, or optional because they may not be available in certain cases; see requirement levels in 7.7.1.

7.3 Recommended HWID tag data values

In addition to the minimum data attributes listed in minimum HWID tag data required (7.2), whenever possible the following data elements should be provided as well:

— `hwidMeta`

— `hwType`

— `modelName`

— `modelNumber`

— `serialNumber`

7.4 XML and JSON naming conventions

7.4.1 XML considerations

HWID schema content shall be identified in accordance with the element and attribute names specified in the data definition subclauses below. This naming requirement ensures consistent interoperability of HWID schema content, regardless of the creator or consumer of the HWID tag data.

XML Schema Definition Language (XSD) 1.1 Part 1: Structures, W3C Recommendation,

XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation,

XML Signature Syntax and Processing Version 1.1, W3C Recommendation 11 April 2013

Elements and types defined in this specification are pascal cased (ThisIsAnExample). (Common simple types like 'string' and 'boolean' are lower case, however.)

Attributes are camel cased (anotherExampleIsLikeThis).

Elements and attributes start and end with a character “a-z”.

Elements and attributes contain only ASCII alphanumeric characters (i.e., “a-z”, “A-Z”, and “0-9”).

Data values in the HWID may be structured as XML elements (which contain XML attributes), XML complex types (which contain elements and/or attributes), or XML attributes (which contain the actual data values). Simple type definitions typically are used to specify sets of acceptable data values.

Example:

```
<ElementExample1>
      attributeExample1=”value1”
      attributeExample2=”value2”
</ElementExample1>
```

7.4.2 JSON considerations

HWID schema content shall be identified in accordance with the element and attribute names specified in the data definitions described in Table 5 to Table 12. This naming requirement ensures consistent interoperability of HWID schema content, regardless of the creator or consumer of the HWID tag data.

Attributes and Elements are Camel cased (anotherExampleIsLikeThis).

Elements and Attributes start and end with a character “a-z”.

Elements and Attributes contain only ASCII alphanumeric characters (i.e., “a-z”, “A-Z”, and “0-9”).

Example:

```
“elementExample1”:{
      “attributeExample1”: “value1”,
```

```
        "attributeExample2": "value2"
}
```

For further information on proper formatting of JSON structures, refer to RFC 7159.

7.5 Language functionality

The language used in this document and all examples is English.

In the XSD there is a "BaseElement" element which allows the definition of a `<lang>` attribute to define the language to be used in any attribute in the HWID.

7.6 Element structure

The logical structure of a HWID is simple. This subclause describes the structure in brief, focusing on the structure of the HWID file.

The most important element in the HWID file is called "HWID".

One or more "entity" elements exist within "HWID", to identify organizations who fulfil certain roles e.g. 'hwidCreator', 'hardwareManufacturer', etc.

Each "HWID" has one or more "hwidMeta" elements associated with it. "hwidMeta" encapsulates the key meta data about the hardware and may include information relating to the contract which applies (which should be described in an ISO/IEC 19770-3 compliant Ent which in turn references the hwidID of this HWID tag).

If the HWID is linked to other HWIDs, especially in the definition of HWIDs of type "System" or "Supplemental", one or more "link" elements exist within "HWID" to identify additional HWIDs this HWID is associated with. In addition, one or more optional external links may also be included within a "HWID".

7.7 Data definitions

7.7.1 Requirement levels

Elements and attributes are marked with a requirement level – the requirement levels are shown in Table 4.

Table 4 — Requirement levels

Requirement level	Definition
M1	Mandatory in all HWIDs. Required for a HWID to be considered complete.
M2	Mandatory in the context of the element. (Used for Mandatory components of non-mandatory elements)
O1	Optional but recommended. If the data is available to the HWID creator, this should be populated.
O2	Optional. This contains information that may be beneficial to the user of the HWID, but not necessary to convey the Hardware details.

HWIDs which have elements which do not adhere to M1 level requirements are considered non-conforming. Non-conforming HWIDs should be flagged by HWID tools, notifying the end-user that these records do not conform to the requirements of this document.

7.7.2 HWID

Table 5 — HWID element description

Element Name	HWID		
Description	Represents the root element specifying data about hardware		
attribute -or- Element	**Type / Multiplicity**	**Require-ment Level**	**Definition**
hwidID	string / 1	M1	hwidID shall be a GUID. The `<hwidID>` provides a globally unique reference for this specific hardware instance. If two HWIDids match and the hwidCreator is the same, the underlying HWID tags they represent are expected to be the same. See 5.2 for details of `<hwidID>` structure.
hwidType	HWIDType / 1	M1	Indicates what the primary purpose of this HWID tag is. See 5.4 above.
supplementalHWIDType	SupplementalHWIDType/ 1	M2	Mandatory if the `<hwidType>` is “Supplemental”. An indicator of the purpose of this supplemental HWID. e.g. To update or archive the linked primary Tag. See 5.5 for suggested types.
hwidCreationDate	dateTimeStamp / 1	M1	The date and time when the HWID was initially generated.
trustLevel	TrustLevel / 1	O2	See TrustLevel in 7.8.9.
sequentialTransactionId	string/1	O2	This is an optional free-format attribute which can be used by HWID creators if they wish to implement transaction sequential counting to ensure the completeness of HWID information transmitted to HWID consumers. This specification does not address how this attribute is to be used. If additional attributes are needed as part of such desired functionality, it is recommended that they be added to the “Meta” sub element and that they refer to this `<sequentialTransactionId>`.
Entity	Entity / 1 - unlimited	M1 for hwid-Creator, O1 for all other roles.	Element that specifies the organizations related to this HWID. This has a minOccurs of 1 because there shall be at least an Entity with `role='hwidCreator'`.
HWIDMeta	HWIDMeta / 1	M1	The key meta data relating to the HWID – includes product and right info (see 7.7.3).
Meta	Meta / 1	O2	An open-ended collection of elements that can be arbitrary metadata about this element. The data provided here should relate to this specific element.
Link	Link / unlimited	O2	Element to reference an external file, a web-based URL or another part of this HWID. The link in this element should point at a specific piece of data which provides additional data relating to this element.

7.7.3 HWIDMeta

Table 6 — HWIDMeta element description

Element Name	HWIDMeta		
Description	"hwidMeta" encapsulates the key meta data about the hardware. "hwidMeta" typically has one or more "Location" elements providing location information for various points of the hardware's lifecycle. 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 HWID, though industry norms for new attributes should be defined and followed to the degree possible. NOTE: The processContents declaration for additional Meta attributes is defined as lax.		
attribute -or- Element	**Type / Multiplicity**	**Requirement Level**	**Definition**
hwType	HWType / 1	M1	Broad classification of hardware type. Example values: "Laptop Computer", "Server Computer", "Printer", "Network Router", "Cable", "Network Switch", etc.
hwSubType	string / 1	O2	More specific classification of hardware type. Example Values: "Laser printer"; "Desktop Switch", "Optical Cable", etc.
hwName	string / 1	O2	End user provided name of the hardware.
isOEM	boolean / 1	O2	To indicate that this part of the HWID is an OEM piece of hardware. This would be true for hardware rebadged by a manufacturer who has an OEM agreement with another manufacturer allowing one to rebrand the others hardware.
isVirtual	boolean/1	M2	Mandatory if this is a HWID for a virtual device.
channelType	ChannelType / 1	O2	Provides information on which channel was used to deliver this HWID. Examples of possible channels can be direct, VAR or OEM.
summary	string / 1	O1	A short (one-sentence) description of the hardware.
description	string / 1	O2	A longer, detailed description of the hardware. This description can be multiple sentences (differentiated from "summary" which is a very short, one-sentence description).
modelName	string / 1	O1	The base name of the product of this type from this manufacturer
modelNumber	string / 1	O1	The number associated with this model, (e.g., 15, 580 G2, 2210).
productSKU	string / 1	O1	A stock keeping unit (SKU) is a unique identifying number for products, as assigned by a hardware provider. The SKU may be represented as a combination of numbers, letters, or symbols. SKU is a commonly used unique number assigned for identification of a piece of hardware for purchase.
productFamily	string / 1	O2	The overall product family the hardware encompassed by this HWID belongs to.

Table 6 *(continued)*

unspscCode	string / 1	O2	An 8-digit code that provides UNSPSC classification of the software product this HWID identifies.
unspscVersion	string / 1	O2	The version of the UNSPSC code used to define the UNSPSC code value.
serialNumber	string / 1	O1	The manufacturer provided serial number of the hardware or virtual hardware defined by this HWID.
cpuManufacturer	string / 1	O2	The manufacturer of the CPU used in this HWID.
cpuCount	string / 1	O2	The number of CPU cores associated with this HWID.
cpuType	string / 1	O2	The CPU model number or description of the CPU associated with this HWID.
keywords	string / 1	O2	This attribute provides the ability for a HWID creator to add a comma separated list of specific keywords to the HWID. The keyword values are not specified in this standard but are instead provided as a way for the HWID creator to help search engines find HWIDs that relate to a particular subject.
assetID	string / 1	O2	The manufacturer provided unique identifier of the hardware or virtual hardware defined by this HWID in addition to the serial number.
Location	Location / unlimited	O2	Location information about this entry. The location can represent the location of an entity as well as location information about the hardware itself as it is installed, stored, or otherwise occupies a physical space. It would typically be included in a Supplemental HWID of supplementalHWIDType = 'AddInfo'. When movements occur, it may be necessary to revoke the old Supplemental HWID where location was stored. All attributes are optional, and only those needed should be used.
OrderInfo	OrderInfo / unlimited	O2	Order information about this hardware. This type of information is not required, but the capability is provided for end-customer organizations which wish to record it. It would typically be included in a Supplemental HWID of supplementalHWIDType = 'AddInfo'. All attributes are optional, and only those needed should be used.
Meta	Meta / 1	O2	An open-ended collection of elements that can be arbitrary metadata about this element. The data provided here should relate to this specific element. Hostname, MAC/IP addresses or FQDN can be examples of Meta fields.
Link	Link / unlimited	O2	Element to reference an external file, a web-based URL or another part of this HWID. The link in this element should point at a specific piece of data which provides additional data relating to this element.

7.7.4 Entity

Table 7 — Entity element description

Element Name	Entity		
Description	Specifies the organizations related to the hardware referenced by this HWID.		
attribute -or- Element	**Type / Multiplicity**	**Requirement Level**	**Definition**
name	string / 1	M1 for hwidCreator, M2 for all others.	The name of the organization claiming a particular role in the HWID. The name of the hwidCreator is mandatory (M1).
regid	string / 1	M1 for hwidCreator, M2 for all others	The regid provides a unique naming authority identifier for the organization. If the regid is unknown, the value “invalid.unavailable” should be used (see IETF RFC 6761 for more details on the default value). See 5.7.2.2 for further discussion on the structure of the regid.
alias	string / unlimited	O2	Other names under which the Entity is known.
role	Role / unlimited	M1 for hwidCreator role; M2 for all others	The relationship between this organization and this HWID e.g. hwidCreator, licensor, etc. The role of hwidCreator is required for every HWID. Role may include any role value, but the pre-defined roles include: — distributor — hardwareManufacturer — hwidCreator Other roles will be defined as real-world experience of using HWIDs is gained.
thumbprint	string / 1	O1	If the HWID is signed, this value provides a hexadecimal string that contains a hash (or thumbprint) of the HWIDities certificate. This allows the digital signature to be directly related to the Entity specified. The HWID Entity element that includes thumbprint shall be included in the signature for the relationship to have any validity.
Location	Location / unlimited	O1	Location information about this entry. The location can represent the location of an Entity as well as location information about the hardware itself as it is installed, stored, or otherwise occupies a physical space.
Meta	Meta / 1	O2	An open-ended collection of elements that can be arbitrary metadata about this element. The data provided here should relate to this specific element.
Link	Link / unlimited	O2	Element to reference an external file, a web-based URL or another part of this HWID. The link in this element should point at a specific piece of data which provides additional data relating to this element.

7.7.5 Link

Table 8 — Link element description

Element Name	Link		
Description	A reference to any another item (can include details that are related to the HWID such as details on product lifecycle, order records).		
attribute -or- Element	Type / Multiplicity	Requirement Level	Definition
linkedToPrimary-hwidID	string /1	M2	Mandatory if the <hwidType> is “Supplemental”, to identify what primary HWID this HWID supplements. Specify the <hwidID> of the HWID to which this HWID is associated here. NIC Cards, RAM, Peripherals, Sensors, etc. are examples of hardware types, which can be associated with another system HWID such as a PC, Router or IOT device. Either linkedToPrimaryhwidID or linkedtoSystemhwidID shall be included for all supplemental HWIDs. Only one of the two attributes should be used. Mandatory attribute for all system HWIDs.
linkedtoSystemh-widID	string /1	M2	Specify the <hwidID> of the HWID to which this HWID is associated here. Primarily used to update or archive system HWIDs. Either linkedToPrimaryhwidID or linkedtoSystemhwidID shall be included for all supplemental tags. Only one of the two attributes should be used.
href	anyURI	M2	A link to the external item being referenced. The href can point to many different things, and can be any of the following: A relative URI (no scheme) - which is interpreted depending on context (e.g. ./folder/supplemental.HWID").. A physical file location with any system-acceptable URI scheme (e.g. file:// http:// https:// ftp:// ...). A URI with "hwid:..." as the scheme, which refers to another HWID by HWIDId. this URI would need to be resolved in the context of the system by software that can look up other HWIDs (e.g. "ent:2df9de35-0aff-4a86-ace6-f7dddd1ade4c"). Only one href attribute is allowed per Link element, however multiple Link elements can be included to reference multiple URI’s.
linkContentId	string / 1	O2	The ID of the downloaded content of a link, as stored in the element LinkContent. This is the key to access content which has been downloaded from the href value given above, e.g., the downloaded content of a link which gives licensing terms and conditions. It is referenced via a key to avoid embedding extensive content repeatedly in different HWIDs.
note	string / unlimited	O2	Free-format note about the link.
rel	Rel / 1	M2	The relationship between this element and the linked item. Relationships can be identified by referencing the IANA Link Relations registration library.
credentials	string / 1	O2	Free text entry of information related to credentials, if needed to access the cited link. For example, this can be the userId and password, comma separated.
type	String/ MediaType	O2	The IANA MediaType for the target href; this provides the HWID tag consumer an indicator of the resource type being referenced. See Internet Assigned Numbers Authority (IANA) for more details on link type.

7.7.6 LinkContent

Table 9 — LinkContent element description

Element Name	LinkContent		
Definition	Provides the actual content which is pointed to by a link, e.g. the downloaded contents of a URI which points to on-line licensing terms and conditions.		
Name	**Type / Multiplicity**	**Requirement Level**	**Definition**
linkContentId	string / 1	M2	An identifier which at a minimum should be unique in the context of this HWID. Incremental numbers are possible. Optionally, an identifier which is unique in a broader context, at least for the end-user organization. Potentially a GUID. Such a value allows for potential normalization of different link contents, for referencing from multiple HWIDs.
linkContent	string / 1	M2	A copy of the linked content.

7.7.7 Meta

Table 10 — Meta element description

Element Name	Meta		
Definition	An open-ended collection of elements that can be arbitrary metadata about the containing element. The data provided here should relate to this specific element. Permits any user-defined attributes in Meta elements.		
attribute - or - Element	**Type / Multiplicity**	**Requirement Level**	**Definition**
<any>	<any>	O2	<any>

7.7.8 OrderInfo

Table 11 — OrderInfo element description

Element Name	OrderInfo		
Definition	Order information about this hardware. This type of information is not required, but the capability is provided for end-customer organizations which wish to record it. It would typically be included in a supplemental HWID of `supplementalHWIDType = 'AddInfo'`. All attributes are optional, and only those needed should be used.		
Name	**Type / Multiplicity**	**Requirement Level**	**Definition**
internalOrderNo	string / 1	O2	End-user organization's internal order number, e.g., requisition number or work order number.
internalOrderDate	date / 1	O2	Date of internalOrder.
purchaseOrderNo	string / 1	O2	End-user organization's purchase order number to its direct supplier, whether this is a reseller or the licensor directly.
purchaseOrderDate	date / 1	O2	Date of purchaseOrder.
purchaseOrderLineItemNo	integer / 1	O2	Line item number in the above, for tracking and reconciliation purposes.
directSupplierPONo	string / 1	O2	If the hardware is obtained indirectly via a reseller (who is therefore the direct supplier to the end-user organization), then the reseller's PO number to the licensor may be important for tracking and reconciliation purposes.
directSupplierPODate	date / 1	O2	Date of directSupplierPONo..

Table 11 *(continued)*

directSupplierPOLineItemNo	integer / 1	O2	Line-item number for directSupplierPONo, for tracking and reconciliation purposes.
directSupplierDescription	string / 1	O2	Description for this line item in the supplier PO.
currency	string / 1	O2	Invoice currency.
unitCost	decimal / 1	O2	Unit cost per asset.
totalCost	decimal / 1	O2	Total cost of all assets.
costCenter	string / 1	O2	Coding for end-user organization's cost charging purposes.
purchaseCondition	purchaseCondition/1	O2	An indication of the condition of an asset at time of purchase on this order. — New — Refurbished — Used
Meta	Meta / 1	O2	An open-ended collection of elements that can be arbitrary metadata about this element. The data provided here should relate to this specific element.
Link	Link / unlimited	O2	Element to reference an external file, a web- based URL or another part of this HWID. The link in this element should point at a specific piece of data which provides additional data relating to this element.

7.7.9 Location

Table 12 — Location element description

Element Name	Location		
Definition	Location information about this hardware. This type of information is not required, but the capability is provided for end-customer organizations which wish to record it. It would typically be included in a supplemental HWID of `supplementalHWIDType = 'AddInfo'`. All attributes except locationType are optional, and only those needed should be used.		
Name	**Type / Multiplicity**	**Requirement Level**	**Definition**
locationType	string / 1	M2	Type of this Location element to denote various locations the hardware. As hardware can move locations over its lifecycle, a device can have a purchasedLocation at address 1, but an installedLocation at address 2. Common types include (but are not limited to): — shipToLocation — installedLocation — taxableLocation — virtual — <any>
installDate	datetime / 1	O2	Date and time device was installed at this location.
removalDate	datetime / 1	O2	Date and time device was removed from this location.
address	string / 1	O2	Address of the entity. This can be in any format with any number of identifiers. This can be useful to identify the particular location for multi-location organizations.

Table 12 *(continued)*

postalCode	string / 1	O2	Post code or zip code of the entity.
city	string / 1	O2	The name of the city associated with the entity.
state	string / 1	O2	The name of the state or region associated with the entity.
country	string / 1	O2	The country with which the entity is associated. This should be an Alpha-2 code from ISO 3166-1.
siteID	string / 1	O2	Site ID associated with the entity. This optional field allows for end users to associate site designators to the HWID.
building	string / 1	O2	Optional field to store building number or name.
room	string / 1	O2	Optional field to storeroom number or name.
rack	string / 1	O2	Optional field to store rack number or name.
bin	string / 1	O2	Optional field to store storage bin number or name.
slot	string / 1	O2	Optional field to store storage slot number or name.
Meta	Meta / 1	O2	An open-ended collection of elements that can be arbitrary metadata about this element. The data provided here should relate to this specific element.
Link	Link / unlimited	O2	Element to reference an external file, a web-based URL or another part of this HWID. The Link in this element should point at a specific piece of data which provides additional data relating to this element.

7.8 Attribute value definitions

7.8.1 General

Several of the elements above require attributes with specific values. The valid values for the attributes are outlined in Tables 13 to Table 21.

7.8.2 ChannelType

Table 13 — ChannelType string enumerated values

Name	ChannelType
Data Type	String
Definition	Provides information on which channel was used to deliver this hardware.
Recommended Values	**Meaning**
Direct	Sold directly from the hardware manufacturer to the end-user organization.
VAR	Sold by a value-added reseller (VAR).
OEM	Sold by an original equipment manufacturer (OEM).
Distributor	Sold by a distributor.
<any>	Additional values are allowed so that HWID licensors can use any `<ChannelType>` description wished.

7.8.3 HWIDType

Table 14 — HWIDType string enumerated values

Name	HWIDType (enumerated values)
Data Type	String
Definition	This allows the differentiation of different types of HWIDs.
Defined Values	**Meaning**
Primary	This is typically as issued by a hardware manufacturer. It defines the basic characteristics of a hardware asset.
System	This is identified by a hardware manufacturer or a HAM system. It defines relationships between different primary components making up a singular system.
Supplemental	This changes an existing HWID by either: — addition of new information not available at the time the primary or system HWIDs were generated. — for the deprecation of existing primary or system HWIDs.

7.8.4 hwType

Table 15 — hwType string enumerated values

Name	hwType
Data Type	String
Definition	Broad classification of hardware type. Different hardware manufacturers may use different terms for similar concepts (note that hwType is not the same as hwidType).
Example Values	**Meaning**
computer server	A computer which is intended to be installed in a data center and/or to provide services to end users or other computers.
desktop computer	A desktop computer intended to be used by one or more users at home or in an office context.
notebook computer	A portable laptop computer intended to be used by one or more users at home or in an office context.
thinclient computer	A desktop computer that has no built-in storage. They connect to a network server and run applications on a remote desktop.
printer	A device for printing pages from computers, either over a network or direct cable connection.
video display	A device for viewing information or content.
handheld device	A tablet or other handheld device.
peripheral	A generic device attached to another device.
scanner	An input device that scans documents such as photographs and pages of text. This creates an electronic version of the document that can be viewed and edited.
data storage	A device to store data for one or more other systems which might include disk, tape, optical or other technologies.
power conditioner	A device to provide clean uninterrupted power input.
network switch	A network device which enabled computers to connect to other computers in a local area network.
network router	A network device which connects two or more disparate networks.
interface card	A daughter card which adds additional capability, functionality, or interfaces.
firewall	A device for limiting or blocking access to other networked devices.
VPN concentrator	A type of networking device that provides secure creation of VPN connections and delivery of messages between VPN nodes.
camera	A device for recording visual images.

Table 15 *(continued)*

sensor	A device which detects or measures a physical property and records, indicates, or otherwise responds to it.
wireless access point	A networking hardware device that allows other Wi-Fi devices to connect to a wired network.
wireless controller	A device used in to manage light-weight access points in large quantities by the network administrator or network operations center.
IOT device	A device deployed for the purpose of connecting and exchanging data with other devices and systems over the internet.
virtual machine	A computer system created using software on one physical computer in order to emulate the functionality of another separate physical computer.
rack	A structure for holding or mounting IT assets.
cable	Used to connect two or more devices for the passing of signals
desktop phone	A device for device that permits two or more users to conduct a conversation.
mobile phone	A handheld, portable device that permits two or more users to conduct a conversation.
<any>	Additional values are allowed so that hardware manufacturers can use any `<hwType>` description they desire. It is recommended that wherever possible the value should align with UNSPSC code "Title" fields for the sake of greater consistency.

7.8.5 purchaseCondition

Table 16 — purchaseCondition string enumerated values

Name	purchaseCondition
Data Type	String
Definition	This attribute represents the condition a device was in at the time of purchase to indicate if it was acquired new from the manufacturer or if it had a previous owner. This applies to supplemental OrderInfo, and not to the primary HWID data itself.
Example Values	**Meaning**
New	A hardware asset that is purchased new in an untouched state from the OEM.
Refurbished	A hardware asset that has been used by previous owners but has undergone maintenance to restore it to a 'like new' state.
Used or Recycled	A hardware asset that is purchased from a previous owner.

7.8.6 LocationType

Table 17 — LocationType string enumerated values

Name	LocationType
Data Type	String
Definition	Broad classification of the type of location being described. Multiple locations may be of interest to the end-user within a HWID including location of manufacture, location of shipment, location of installation, or location of storage.
Example Values	**Meaning**
shipToLocation	Location that the asset represented by this HWID was shipped to. This location may not represent where it was stored or installed.
installedLocation	Location that the asset represented by this HWID is installed. Multiple installedLocations may be defined for hardware that is moved between locations, with uninstalled devices having the removalDate attribute defined.

Table 17 *(continued)*

taxableLocation	Location that the asset represented by this HWID is considered being located for tax purposes. This location may not represent where it was originally shipped to.
storageLocation	Location that the asset represented by this HWID is stored. Multiple storageLocations may be defined for hardware that is moved between locations, with uninstalled devices having the removalDate attribute defined.
<any>	Any other locationType may be defined so that HWID creators can properly define the purpose of a location.

7.8.7 Role

Table 18 — Role string enumerated values

Name	Role
Data Type	String
Definition	Specifies how an Entity relates to this HWID. One Entity may serve many roles, e.g., as hardwareManufacturer, licensor, and hwidCreator. Pre-defined values are recommended for use, but it is expected that new roles may be required as market needs change. This type includes the ability to define additional values as required.
Recommended Values	**Meaning**
hwidCreator	HWID creator
trustAssigner	Entity which established the TrustLevel for this WHID (See TrustLevel in 7.8.9).
entitledEntity	Entity which can use/has ownership of the hardware described in this HWID.
directSupplier	Entity which has sold or otherwise supplied, to the end-user organization, the entitlement described in this HWID.
distributor	An entity that furthers the marketing, selling and/or distribution of software from the original place of manufacture to the ultimate user without modifying the software, its packaging, or its labelling.
licensor	Software licensor (see ISO/IEC 19770-5).
hardwareManufacturer	Hardware creator
<any>	Additional values are allowed so that HWID creators can extend the roles they describe.

7.8.8 SupplementalHWIDType

Table 19 — SupplementalHWIDType string enumerated values

Name	SupplementalHWIDType
Data Type	String
Definition	This allows the differentiation of different types of supplemental HWIDs.
Recommended Values	**Meaning**
InfoAdded	This supplemental record is to add information (but not quantities) to an existing HWID. If different data values are added for the same element or attribute (whether in the same or in different supplemental HWIDs), then the interpretation of that data, i.e., the order of precedence, is not specified by this standard. (supplemental HWIDs with incorrect or changed values should be revoked).

Table 19 *(continued)*

Archived	This supplemental HWID archives an existing HWID. The <hwidID> for the HWID which is archived is specified in the <linkedToPrimaryhwidID> or <linkedToSystemhwidID> attribute. If the <linkedToPrimaryhwidID> or <linkedToSystemhwidID> value indicates a HWID which has the <hwidType> of "Supplemental" then only that supplemental HWID is archived. If the <hwidID> indicated in the <linkedToPrimaryhwidID> or <linkedToSystemhwidID> attribute is not of <hwidType> "Supplemental" then that <hwidID> and all supplemental HWIDs associated with that <hwidID> are to be considered archived. The primary reasons that a HWID would be marked archived are: — The HWID referenced needs to be replaced because of incorrect or missing information. Instead of adding the information with an InfoAdded, the entire HWID is archived and reissued. — The hardware or system described by the HWID referenced has been removed from inventory, and therefore no longer to be actively used for hardware management purposes. This would be appropriate, for example, for hardware resold, destroyed or is otherwise permanently unavailable. Note that hardware that is transferred to another entity might be better served by updating the entity information of the new owner to maintain history and continuity of the HWID tag.
	The effect of the 'Archived' supplemental HWID is the same as that of 'Revocation', but with an identified purpose.
<any>	Any other value of may be used, to provide extensibility in this document for additional uses not yet specified.

7.8.9 TrustLevel

Table 20 — TrustLevel string enumerated values

Name	TrustLevel (enumerated values)
Data Type	String
Definition	The trust level assigned to a HWID. This is effectively a subjective value, with the highest level of trust generally being assigned to a HWID created by the licensor.
Defined Values	**Meaning**
Full	Full trust, e.g., for a HWID created by a hardware manufacturer.
Partial	Partial trust, e.g., created by an end-user organization on the basis of historical HWID records from the publisher.
Untrusted	Data is not reliable, e.g., based on records from resellers rather than from the licensor; or based on other non-original sources.

7.8.10 Rel

Table 21 — rel string enumerated values

Name	rel (enumerated values)
Data Type	String
Definition	Specifies how the target of a [Link] relates to this HWID.
Defined Values	**Meaning**
primary	A reference to a primary HWID tag for the purpose of linking HWIDs together to create a system. May also be used link together system of systems.
ancestor	Refers to a HWID tag defining an ancestor of this HWID; used for upgrades.
parent	A reference to a HWID tag that this HWID is a child of.
child	A reference to a HWID tag that this HWID is a parent of.

Table 21 *(continued)*

lifecycle	A reference to additional information on lifecycle characteristics of this hardware product (end of support, end of life, etc.).
supplemental	Additional information that represents *this* HWID. This may be used to provide additional information for a hardware product such as the purpose for the installation.
contract	Refers to a URI where a copy of the hardware's relevant contract information can be found.
bundleLeadProduct	The link is to another HWID which is for the lead product in a group of products sold as a bundle. (The current HWID is not for the lead product in the bundle).
bundledWith	The link is to another HWID which is for a product bundled together with this product. (The current HWID is for the lead product in the bundle).
<any>	Additional relationships can be identified by referencing the IANA Link Relations registration library.

Annex A
(informative)

XML schema definition (XSD)

The following XSD document provides the definition for the hardware schema.

```
      <?xml version="1.0" encoding="UTF-8"?>
  <xs:element name="HWID">
    <xs:complexType mixed="true">
      <xs:sequence>
        <xs:element name="Entity" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType mixed="true">
            <xs:sequence>
              <xs:element type="xs:string" name="Location" maxOccurs="unbounded"
minOccurs="0"/>
                <xs:sequence>
                  <xs:element type="xs:string" name="Meta" maxOccurs="unbounded"
minOccurs="0"/>
                  <xs:element type="xs:string" name="Link" maxOccurs="unbounded"
minOccurs="0"/>
                </xs:sequence>
              <xs:element type="xs:string" name="Meta" maxOccurs="unbounded" minOccurs="0"/>
              <xs:element type="xs:string" name="Link" maxOccurs="unbounded" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="hwidMeta">
          <xs:complexType mixed="true">
            <xs:sequence>
              <xs:element type="xs:string" name="Location" maxOccurs="unbounded"
minOccurs="0"/>
                <xs:sequence>
                  <xs:element type="xs:string" name="Meta" maxOccurs="unbounded"
minOccurs="0"/>
                  <xs:element type="xs:string" name="Link" maxOccurs="unbounded"
minOccurs="0"/>
                </xs:sequence>
              <xs:element type="xs:string" name="OrderInfo" maxOccurs="1" minOccurs="0"/>
                <xs:sequence>
                  <xs:element type="xs:string" name="Meta" maxOccurs="unbounded"
minOccurs="0"/>
                  <xs:element type="xs:string" name="Link" maxOccurs="unbounded"
minOccurs="0"/>
                </xs:sequence>
              <xs:element type="xs:string" name="Meta" maxOccurs="unbounded" minOccurs="0"/>
              <xs:element type="xs:string" name="Link" maxOccurs="unbounded" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element type="xs:string" name="Meta" maxOccurs="unbounded" minOccurs="0"/>
        <xs:element type="xs:string" name="Link" maxOccurs="unbounded" minOccurs="0"/>
        <xs:element type="xs:string" name="LinkContent" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```