International Standard

ISO/IEC 18975

First edition
2024-11

Information technology — Automatic identification and data capture techniques — Encoding and resolving identifiers over HTTP

Technologies de l'information — Techniques automatiques d'identification et de saisie de données — Encodage et résolution des identifiants via HTTP

Reference number
ISO/IEC 18975:2024(en)

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*, SC 31, *Automatic identification and data capture techniques*.

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

There are many identifier systems in use today, some of which enjoy widespread usage and long histories. Examples include the International Air Travel Association (IATA) airport codes and identifiers such as digital object identifiers (DOIs) (see ISO 26324[5]), and country and currency codes. The ISO/IEC 15459 series[7] provides the basis for all identifier issuing agencies in the field of Automatic Identification and Data Capture (AIDC).

The ISO/IEC 15459 series[7] defines methods for ensuring that identifiers are globally unique in the world of AIDC without any reliance on, for example, the internet's domain name system or any given data service. This is critical for use cases where internet connectivity and the availability of online services cannot be allowed to affect whether a process can be completed, such as the purchase of an item.

However, existing identifiers can be usefully encoded in Hypertext Transfer Protocol Uniform Resource Identifiers (HTTP URIs) following Linked Data principles so that when connectivity is available, they can be used in multiple methods of online lookup, data query and data integration.

It is important to note that identity on the internet is defined by the domain name system. The Internet Engineering Task Force (IETF)'s Best Current Practice[2] makes it clear that each domain is sovereign over the URIs under that domain. This document brings the internet, Linked Data[8] and AIDC together to allow the discovery of online data related to physical objects in a way that recognizes and respects both approaches to globally unique identity. It further defines a common approach to how those URIs can be associated with links to multiple sources of data in addition to the one encoded directly in the HTTP URI.

This document focuses primarily on environments in which the URI is parsed offline to extract identifiers that are globally unique in their own right, irrespective of the internet domain name used. This applies especially, but not only, to identifiers that conform to the ISO/IEC 15459 series[7]. For use cases and environments where it is appropriate to rely on the internet domain name to confer global uniqueness on some or all aspects of identification, the IEC 61406 series[17][18] is likely to be relevant, especially for technical/engineering industries.

Information technology — Automatic identification and data capture techniques — Encoding and resolving identifiers over HTTP

1 Scope

This document specifies the different approaches for using HTTP URIs to encode globally unique identifiers. It specifies a dual use data structure. It is both an HTTP URI and a composition of structured item identification properties and optionally descriptive attributes. These can be decomposed and interpreted on their own and/or be used as a pointer to additional information.

Methods are described to enable identification uniqueness in the context of AIDC. These rely on either:

a) identifiers, such as described in the ISO/IEC 15459 series[7], in the path or query string independent of the internet domain name; or,

b) the internet domain name.

The document further defines a basic common API for querying online services for information about identified items.

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 19762, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax. T Berners-Lee, R Fielding, L Masinter. IETF 2005*

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 19762 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
dereference
look something up on the internet

Note 1 to entry: For a full definition, see RFC 3986.

3.2 Abbreviated terms

AI	Application Identifier, as defined in ISO/IEC 15418[14]
AIDC	Automatic Identification and Data Capture (barcodes, RFID tags, etc.)
API	Application Programming Interface
CURIE	Compact URI (defined in Reference [4])
DI	Data Identifier, as defined in ISO/IEC 15418[14]
GTIN	Global Trade Item Number
HTTP	Hypertext Transfer Protocol (the underlying internet communication protocol for the World Wide Web)
HTTP URI	equivalent to the better-known term “URL” (Uniform Resource Location) but in this context, the term URI is more appropriate.
IANA	Internet Assigned Numbers Authority
JSON	JavaScript Notation Language – a data structure commonly used in online data exchange (defined in ISO/IEC 21778[15])
URI	Uniform Resource Identifier (defined in RFC 3986)

4 URI structures

4.1 Framework

Subclauses 4.2 and 4.3 define two frameworks on top of which more precise definitions may be added to create rules for encoding specific types of identifier in an HTTP URI. This document does not define a complete system that can be implemented directly without the additional detail provided in other standards and guidelines.

4.2 Structured path approach

The structured path approach makes a clear distinction between strings of characters that are identifiers and those that are descriptive attributes. Identifiers, such as the item’s class identifier and serial number, are encoded in an ordered sequence in the path. Descriptive attributes such as the mass or size of an item are not part of the identification of the item and are encoded in the query string as name=value pairs.

In many cases, the descriptive attributes are aligned with the identifiers and may be retrieved through a lookup of the HTTP URI. For example, the expiration date of a perishable product may be known by looking up its product identifier and the identifier of the batch in which it was produced. The presence of such descriptive attributes in the HTTP URI is a convenience designed to eliminate the need for a lookup in high-speed environments, and the HTTP URI will function equally well without it in most use cases.

Applying this logic to encoding the ISO/IEC 15459 series[7] identifiers in HTTP URIs leads to a design principle for the structured path approach, as follows:

— Identifiers shall be encoded in path segments in order of increasing granularity from left to right.

— Descriptive attributes shall be encoded, in any order, in the HTTP URI’s query string.

At a high level, this creates the structure shown in Figure 1 in which the structure itself is part of the data payload.

https://example.com/class-id/sub-class-id/instance-id?description$_1$=value$_1$&description$_2$=value$_2$

NOTE The example.com domain name is used as defined in RFC 6761[10].

Figure 1 — High-level view of a URI following the structured path approach

Any identifier issued by an Issuing Agency conforming to the ISO/IEC 15459 series[7] will come with information about the identifier's qualifier. These can be considered as parameter names for which the identifiers themselves are the values. The identifier qualifiers are an important part of the data payload whether encoded in a URI or some other syntax. Bearing this in mind, a more detailed version of the path segments for the example in Figure 1 is as shown in Figure 2.

https://example.com/(pathSegments/)$_n$primaryQualifier/primaryID/(qualifer/ID)$_n$?description$_1$=valu e$_1$&description$_2$=value$_2$...

Figure 2 — Detailed view of the path segments of a URI following the structured path approach.

The components of a URI following the structured path approach shown in Figure 2 are as follows:

a) As defined by RFC 3986, all URIs begin with the scheme followed by a colon. HTTP URIs therefore begin with http: or https:. This is followed by a double forward slash (//) and the internet domain name. Optionally, the domain name may be replaced by an IP address and either may be followed by a port number. When constructing an HTTP URI following the structured path approach defined in this document, the port number, username and password shall not be used.

b) The URI may include arbitrary path segments after the internet domain name.

c) For emphasis, those arbitrary path segments and the internet domain name are not part of the identification of the item. They do, however, provide a convenient method for looking up online information about the identified item.

d) Every URI following the structured path approach shall have exactly one primary qualifier and identifier.

e) The primary qualifier, i.e. the type of identifier, is the first path segment that contributes to the identity of the item. This may be either a class-level or instance-level identifier.

f) This is followed by a path segment containing the primary identifier itself.

g) Depending on the type of identifier and any rules defined for its use, primary qualifiers and identifiers may be followed by further path segments that identify sub-classes and/or instances. For example, if the primary qualifier and identifier identify a class of pump, the next two path segments can be a qualifier and identifier for a specific batch of pumps, followed by two further path segments that provide a unique item identifier.

h) The query string comprises 0 or more name=value pairs of qualifiers and data that describe, but do not identify the item. To extend the pump example, they can describe the size or date of production.

i) Although RFC 3986 allows a number of different delimiters to be used in query strings, HTTP URIs conforming to this document shall use the ampersand (&).

j) Reserved characters, as defined by RFC 3986, shall be percent encoded when they appear as identifiers or their values.

An example using the structured path approach is provided in Annex A.

4.3 Query string approach

As with the structured path approach, in the query string approach, the scheme and domain name may be followed by arbitrary path segments. However, all the qualifiers and identifiers, and descriptive attributes, are provided as name=value pairs in the query string, delimited by ampersands (&s). See Figure 3.

$$\text{https://example.com/(pathSegments/)}_n\text{?qualifier}_1\text{=ID}_1\text{(\&qualifer=ID)}_n\text{\&description}_1\text{=value}_1\text{\&description}_2\text{=value}_2\ldots$$

Figure 3 — Generic example of a URI following the query string approach

Qualifiers used as names in the query string may be preceded by one or more characters to indicate that the qualifier and its identifier may be treated as being globally unique in accordance with the rules of the identification system used independent of the internet domain name in the URI. Other parameter names and their values contained in the query string that are not preceded by the same character(s) shall not be treated as part of the system that confers globally unique identification, but only as parameters relevant to the specified internet domain.

An example using the query string approach is provided in Annex A.

4.4 Semantic differences

At a high level, both the structured path and query string approach achieve the same goals. The HTTP URI is a convenient method to find information about the identified item. However, there are important differences in the semantics of the two approaches.

The domain name and path segments of an HTTP URI form a hierarchy that identifies a specific resource on the web. The query string contains name=value pairs that are passed as parameters to that resource. This means that the structured path approach identifies a different online resource for each identified item. Applying Linked Data principles means that information about an item can be inherited from higher up the hierarchy where available. In contrast the query string approach identifies a resource from which information about any number of identified items can be obtained. There is no inherent hierarchy in the organization of the name=value pairs.

Annex C provides background information on the semantic differences between the two approaches.

4.5 Canonicalization

The generic URI syntax (RFC 3986) supports a significant amount of flexibility. For example, the scheme and internet domain name are case insensitive. They both normalize to lower case and many clients will perform this normalization before attempting to dereference the URI. This can allow for more efficient encoding in, for example, QR codes as discussed in Annex B. Paths and query string parameters are case sensitive. It is also noteworthy that any fragment identifier appended to an HTTP URI is not transmitted to the server but may be processed locally.

The semantics of URI query strings are such that the order in which name=value pairs appear is not important. Querying an online resource for $\text{name}_1\text{=value}_1\text{\&name}_2\text{=value}_2$ will return exactly the same result as $\text{name}_2\text{=value}_2\text{\&name}_1\text{=value}_1$.

As URIs are passed from client to server, perhaps via intermediate services, they can change in some respect. Where globally unique identifiers are included, especially those derived from the ISO/IEC 15459 series[7], it is possible for the domain name to be changed completely and query string parameters added or removed without affecting which entity is identified.

Noting this flexibility, in some use cases it may be useful to have a canonical representation of the string that can be generated independently of any individual data system. This can be useful for simple string matching, for example, and is essential in use cases that require the identifier string to be obfuscated through hashing.

As this document acts solely as a framework, it is for application standards and guidelines to define their own canonical form of HTTP URI. Such documents should specify:

a) a specific internet domain name and whether it should be expressed in upper or lower case;

b) a strong preference for https rather than http and, again whether it should be expressed in upper or lower case;

c) that name=value pairs, if present, should be sorted in a defined order;

d) rules for handling elements of the URI that must be treated as immutable, such as the identifiers themselves and the ordering of path segments, and other elements that may be amended or discarded in the canonicalization process;

e) the `linkType` parameter shall not be included (see 5.2);

f) whether or not the canonical form shall include a fragment identifier.

5 Resolution

5.1 Linkset

Application standards and guidelines based on this document are strongly encouraged to support RFC 9264[13]. This allows an identified item to be linked to multiple sources of information about it. Information can be anywhere online, perhaps managed independently by multiple parties. In this way, the identified item itself becomes a central node in a distributed dataset.

5.2 LinkType parameter

This document builds on RFC 9264[13] to define a type of resolver. As well as being able to return a linkset, a conformant resolver will be able to redirect to any of the links in the set associated with the identified item. A request to the resolver is formed by setting the value of the `linkType` parameter to the desired link relation type and appending it to the HTTP URI that identifies the item. The resolver makes a best effort to select a match for the requested link relation type from the links available in the linkset and redirect to it immediately. Link relation types, other than those listed in the IANA registry[9] defined by RFC 8288[12], should be expressed as CURIEs[4].

Annex D provides worked examples of how an identified item can be queried for specific information.

6 Online considerations

All internet domain names have complete sovereignty over their URI space. Applications should not assume that any URI that conforms to either the structured path or query string approach has necessarily been minted for use as described in this document. It is up to individual applications to decide whether the context in which the URI is found is sufficient reason to treat it as anything other than an opaque identifier. This is likely to be the case when encoded in an AIDC data carrier attached to an object, but it cannot be guaranteed.[2]

However, whether a server operates as a resolver can be determined with confidence. Servers that are programmed to respond with a linkset in response to an appropriate query, and to process the `linkType` parameter, shall include a JSON file at `/.well-known/resolver` (the use of /.well-known is defined in RFC 5785[11]). This is known as the resolver description file.

The JSON file can include any data but as a minimum shall include the following:

a) A name for the resolver:

   1) property: `name`

   2) type: string

b) The resolver root:

   1) property: `resolverRoot`

   2) type: URI

c) The namespace(s) of supported link types and associated CURIE prefixes (see 5.2):

   1) property: `supportedLinkType`

   2) type: array of objects

   3) property: `namespace`

   4) type: URI

   5) property: `prefix`

   6) type: string

Annex A
(informative)

Examples

The example shown in Figure A.1 includes a class level identifier and a unique item identifier of a member of that class, together with information about the date of manufacture of that item.

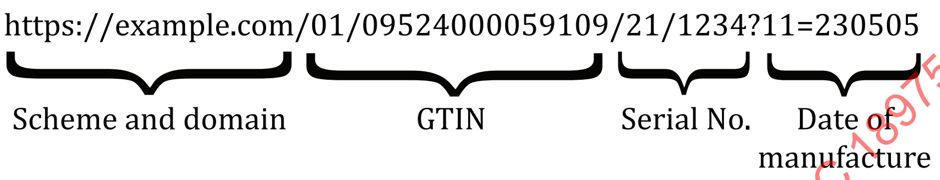

Figure A.1 — An example of a structured path URI that uses GS1 Application Identifiers

The example shown in Figure A.2 identifies the same class of item, but not a unique instance of it. Notice that it is on a different domain name and that there is an extra segment in the path, neither of which contributes to the identification of the item.

These ideas are expanded upon to provide a fully implementable system in GS1 Digital Link URI syntax[6].

https://retailer.example/our-range/01/09524000059109

Figure A.2 — A class-level identifier that is semantically equivalent to that in Figure A.1 despite being on a different internet domain and not including the serial number or date of manufacture

Figure A.3 uses the query string approach. Note the inclusion of the "." character to precede the DIs. This document does not require the inclusion of the "." but application standards and guidelines that build on it may do so. Note that the final name=value pair is not preceded by a dot. As defined in 4.3, this means that foo=bar should only be considered as unique on the example.com domain. If encoded in a data carrier, the full text of the URI as encoded, including the upper case scheme and domain name, is the canonical form of this identifier.

These ideas are expanded upon to provide a fully implementable system in the AutoID specification[1].

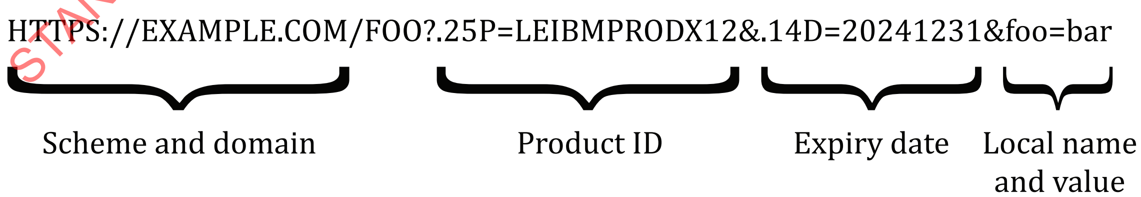

Figure A.3 — An example of the query string approach using ANSI MH10.8.2[16] Data Identifiers to identify a product and its expiry date

Annex B
(informative)

Code size optimization

In almost all situations, it is important to minimize the physical size of an optical data carrier such as a QR code or Data Matrix. The efficiency of encoding used in those symbols decreases from digits, to upper case alphas to lower case alphas and to other characters.

As noted in 4.5, the scheme and domain name are case insensitive. This means that https://EXAMPLE.COM, https://example.com and all mixtures of upper and lower case alphas all dereference to the same point on the web.

Taking these two factors into consideration, it is recommended that upper case alphas be used for the scheme and domain name. The two QR codes shown in Figure B.1 exemplify this. In the online context, they both carry exactly the same data. However, by using upper case alphas, the size of the QR code can be reduced.

https://example.com HTTPS://EXAMPLE.COM

Figure B.1 — Example of QR code encoding size in lower case alphas and upper case alphas

Two further points to note:

a) It is only the scheme and domain name that are case insensitive. All other components of an HTTP URI are case sensitive. The size of the data carrier can be minimized if the data itself comprises digits if possible, then upper case alphas.

b) Many HTTP clients will normalize the domain name to lower case before dereferencing the URI. That is, the string "EXAMPLE.COM" is converted to "example.com" before being passed across the internet. This is an example of the kind of manipulation that can occur on the web as noted in 4.5.

Annex C
(informative)

Background information

This document describes two distinct approaches to encoding identifiers in HTTP URIs.

Both approaches can be used to encode existing identification keys that are globally unique, irrespective of the internet domain name in the URI. This separation between the identifiers on the one hand, and the internet domain name and any service referenced in the URI on the other, is important for use cases where the identifiers are used without any need for an online lookup. The URI does, however, provide one possible and convenient place to look up information about the item if needed.

In the structured path approach, identifiers and any descriptive attributes are treated separately with identifiers being placed in order of increasing granularity in the path of the URI, and descriptive attributes in the query string (in any order). This approach follows Linked Data principles to allow flexible querying methods and data integration. It prioritizes the identification of the object itself as distinct from any digital information about it.

In the second approach, all identifiers and attributes are encoded in the query string as parameters for a query against a resource identified by the internet domain name and path in the URI. Identifiers in the query string can be, but are not always, globally unique in other contexts. Where globally unique identifiers are used in the query string, again, they can be extracted independently of the internet domain name.

Where identifiers are not globally unique in their own right, the complete URI, including its internet domain name, can be used by any system as a unique identifier for the item. It is not intended to be parsed by scanning devices that should simply pass on the complete string.

Of particular importance are identification keys issued by agencies conforming to the ISO/IEC 15459 series[7]. AIDC scanners can recognize and extract those keys and pass them to their host system in exactly the same way as when those same keys are encoded in other syntaxes. From an AIDC perspective, no online lookup is necessary. However, the HTTP URI can also be followed by non-specialist devices, especially smartphones equipped with cameras.

Both approaches provide a general-purpose method of connecting to online information about an identified item and conform to the URI general syntax defined by RFC 3986.

Structured Path URIs are a feature of Linked Data and arise naturally, for example, from the way people typically organize information on their computers, which is the basis for the original design of URIs. The classic example of this is shown in Figure C.1.

https://example.com/

the homepage of Example Inc.

https://example.com/people

the list of staff at Example Inc.

https://example.com/people/alice

information about Example Inc's employee, Alice

https://example.com/people/bob

information about Example Inc's employee, Bob

NOTE Adapted from Reference [3].

Figure C.1 — Example of structured URIs that provide more granular identification in path segments ordered from left to right

The structure reflects concepts that are familiar from information management, namely that of a class, zero or more sub-classes, and instances. Referring to Figure C.1, there will be some facts about Alice and Bob that are unique to those individuals, but they can both “inherit” facts that are common to all members of staff at Example Inc., and all staff inherit facts about the company itself.

Applications can query data sources at the appropriate level of granularity for their needs which may only make use of some elements of the structured path URI. For example, the query may make use of, but equally may ignore, some of the more granular identifiers. This means that there are multiple ways to make use of the HTTP URI, both online and offline, by just following the structure in the path.

Annex D
(informative)

Querying an identified item

An identified item will typically be the subject of multiple sources of information. To revisit an example mentioned in 4.2, an engineering part, such as a pump, can have information that describes its features and specification. This would be provided by the manufacturer. The pump may be used as one part in a more complex system installed by a second company that provides information about the overall system. A third company can be responsible for maintenance of the pump. A change of maintenance contract can mean that, over time, several companies have records for that specific item. Emergency services sometimes need quick access to information about what to do should the pump fail.

In healthcare, different information is typically offered to clinical staff and patients. In retail, the primary distinction is between brand owners and consumers. In all scenarios, the same information can be needed by users who speak different languages, systems that work with different data formats and serializations, etc.

In each of these use cases, it is very likely that information about the item will be created and managed by multiple stakeholders. Some of the information will be commercially sensitive and subject to access control – control that itself can be exercised by different people and organizations.

These multiple sources of information can be identified and pointed to in a "linkset", the concept of which is shown in Table D.1.

Table D.1 — Linkset concept

Link relation type	Target URL	Lang
Product Specification	https://manufacturer.example/en/pump123	en
	https://manufacturer.example/de/pump123	de
	https://manufacturer.example/zh/pump123	zh
Installation Info	https://installer.example?model=pump123&lang=en	
Maintenance Record	https://maintainer.example?cust=123&item=ABC	

In this case, there are three links to the product specification from the manufacturer, one each in English, German and Chinese. Separately, there are links to information about the item's installation history and its maintenance record. No language needs to be given for these as there is only one link of each type.

Note that the target URLs do not conform to this document. They are simple URLs that point to the kind of information described in the link relation type.

The syntax and use of link relation types on the internet are defined in RFC 8288[12]. This establishes a registry of link relation types that can be used without qualification. However, they are unlikely to be useful in implementations of this document. Therefore, the expectation is that the extension mechanism defined in RFC 8288[12] will be used to express link relation types suitable for the context of this document.

Link relation types are defined as URIs that should resolve to their definition. Interoperability is greatly enhanced if well-known link relation types are used across multiple systems. Figure D.1 shows an example of a link set that uses the JSON serialization defined in RFC 9264[13] to encode the example data from Table D.1. Note that the link relation types are defined here using "https://example.net/lrt/" as an example namespace. The use of example.net rather than example.com, as used in the other examples in this document, indicates that the link relation types can be defined anywhere and do not need to be defined by the provider of the link set.