



Guidelines for use of a 32-bit or 40-bit Context Dependent Identifier (CDI-32™ and CDI-40™)

General

The IEEE defined 32-bit context-dependent identifier (CDI-32™) is a concatenation of a 24-bit Organizationally Unique Identifier (OUI) value administered by the IEEE Registration Authority and an 8-bit extension identifier assigned by the organization with that OUI assignment.

The IEEE defined 40-bit context-dependent identifier (CDI-40™) is a concatenation of a 24-bit Organizationally Unique Identifier (OUI) value administered by the IEEE Registration Authority and a 16-bit extension identifier assigned by the organization with that OUI assignment.

The IEEE administers the assignment of 24-bit OUI values. The assignments of these values are public, so that a user of an CDI-32 or CDI-40 value can identify the manufacturer that provided the value[1]. The IEEE/RAC has no control over the assignments of 24-bit extension identifiers and assumes no liability for assignments of duplicate CDI-32 or CDI-40 identifiers assigned by manufacturers.

Context identification

In the case of the IEEE 802 standards, standard 802b defines a schema for allocating ASN.1 Object Identifier arcs for the use of each working group in IEEE 802. This schema establishes the ability of each IEEE 802 Working Group to assign Object Identifier values for standards use. It is recommended that IEEE 802 Working Groups that define context-dependent identifiers allocate an arc under their "root" ASN.1 Object Identifier value, specifically for the purpose of assigning further sub-arcs to allow an Object Identifier "root" value to be assigned to each context-dependent identifier (or perhaps more correctly, to act as an identifier for the identifier's context). By this means, a globally unique ASN.1 Object Identifier value could be derived from a context-dependent identifier value. For example, the "root" ASN.1 Object Identifier value assigned for use by 802.1 is:

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
```

The next arc could be assigned to be used for 32-bit context-dependent identifiers:

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-32-bit-identifiers(X)
```

The next arc could then be used for each particular 32-bit context-dependent identifier (CDI-32) defined by 802.1. For example, in standard 802.1ZZ, there might be a context-dependent identifier called vendorDependent defined, which could be assigned by 802.1 the following Object Identifier root value:

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-32-bit-identifiers(X) vendorDependent(Y)
```

The next two arcs would then be used for the OUI value and the extension identifier value. Hence, the full, globally unique, ASN.1 Object Identifier value corresponding to a single context-dependent identifier value would consist of a sequence of 8 integers, the final two being the OUI and the extension identifier values. The use of the "short form" (just the OUI and the extension identifier) within the defined context would then be equivalent to using the full Object Identifier value.

The same approach, but with distinct arc values to identify the appropriate SDO and standard, could be adopted in other standards where it is considered appropriate to make use of this type of context-dependent identification.

32-bit context-dependent identifier format tutorial

OUI based CDI-32 sequence

A 32-bit context dependent identifier (CDI-32) consists of two components. The 24-bit first portion of this value is the OUI value assigned to the manufacturer by the IEEE Registration Authority. The 8-bit second portion of this identifier is assigned by the manufacturer.

For example, assume that a manufacturer's IEEE-assigned OUI value is AC-DE-48 and the manufacturer-selected extension identifier for a given component is 23. The CDI-32 value generated from these two numbers is AC-DE-48-23.

Some standards specify a CDI-32 by a string of four bytes, labeled here as *cdi*[0] through *cdi*[3], as illustrated below. Although different standards may specify different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-32-bit-identifiers(W) vendorDependent(X)
```

OUI			ext id	field
cdi[0]	cdi[1]	cdi[2]	cdi[3]	order
AC	DE	48	23	byte

Numerical formats

Other standards may specify an CDI-32 to be a numerical value, upon which computations (such as base/bounds or bit selections) can be performed, as illustrated below:

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-32-bit-identifiers(W) vendorDependent(X)
```

ACDE4823₁₆

40-bit context-dependent identifier format tutorial

OUI based CDI-40 sequence

A 40-bit context dependent identifier (CDI-40) consists of two portions. The 24-bit first portion of this value is the OUI value assigned to the manufacturer by the IEEE Registration Authority. The 16-bit second portion of this identifier is assigned by the manufacturer.

For example, assume that a manufacturer's IEEE-assigned OUI value is AC-DE-48 and the manufacturer-selected extension identifier for a given component is 23-48. The CDI-40 value generated from these two numbers is AC-DE-48-23-45.

Some standards specify a CDI-40 by a string of five bytes, labeled here as *cdi*[0] through *cdi*[4], as illustrated below. Although different standards may specify different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-40-bit-identifiers(Y) vendorDependent(Z)
```

OUI-36			ext id		field
cdi[0]	cdi[1]	cdi[2]	cdi[3]	cdi[4]	order
AC	DE	48	23	45	byte

OUI-36 based CDI-40 sequence

A 36-bit Organizationally Unique Identifier (OUI-36) based 40-bit context dependent identifier (CDI-40) consists of two portions. The 36-bit first portion of this value is specified by the OUI-36 value assigned to the manufacturer by the IEEE Registration Authority. The 4-bit second portion of this identifier is assigned by the manufacturer.

For example, assume that a manufacturer's IEEE-assigned OUI-36 value is AC-DE-48-23-4 and the manufacturer-selected extension identifier for a given component is 5. The CDI-40 value generated from these two numbers is AC-DE-48-23-45.

Some standards specify a CDI-40 by a string of five bytes, labeled here as *cdi*[0] through *cdi*[4], as illustrated below. Although different standards may specify different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-40-bit-identifiers(Y) vendorDependent(Z)
```

IAB					ext	field
<i>cdi</i> [0]	<i>cdi</i> [1]	<i>cdi</i> [2]	<i>cdi</i> [3]	<i>cdi</i> [4]	order	
AC	DE	48	23	45		byte

Numerical formats

Other standards may specify an CDI-40 to be a numerical value, upon which computations (such as base/bounds or bit selections) can be performed, as illustrated below:

```
iso(1) std(0) iso8802(8802) ieee802dot1(1)
context-dependent-40-bit-identifiers(Y) vendorDependent(Z)
```

ACDE482345₁₆

Context specifiers within standards

Since a CDI-32 value is only unique within its context, standards are responsible for specifying the context adjacent to the specification of CDI-32 fields or values, as illustrated in preceding [Byte sequence formats](#) and [Numerical formats](#) subclauses. Similarly, since a CDI-40 value is only unique within its context, standards are responsible for specifying the context adjacent to the specification of CDI-40 fields or values, as illustrated in preceding [Byte sequence formats](#) and [Numerical formats](#) subclauses.

If CDI-32 values are provided for distinct applications within one standard, and CDI-32 values are only required to be unique within each application, then distinct CDI-32 arcs should be

specified. Similarly, if CDI-40 values are provided for distinct applications within one standard, and CDI-40 values are only required to be unique within each application, then distinct CDI-40 arcs should be specified.

[1]Except for private OUI values, where the owner of the OUI value is confidential. These remain private.

Copyright © 2005 IEEE

(IEEE Standards Systems/Network Staff)

URL: <http://standards.ieee.org/regauth/oui/tutorials/EUI48.html>

(Modified: 10-March-2005)