IEEE/RAC vote proposal (draft)

1. Voting options

___ Update the IEEE/RAC web site with tutorials for using the IAB with the CDI-40, EUI-48, and EUI-64, as described in attachments.

___ Delete all mention of the IAB, throughout the IEEE/RAC write-ups and discontinue this service.

2. Background

The IAB is a 12-bit extension of the 24-bit OUI that forms a 36-bit number. As per accepted ballots, that 36-bit number can be supplemented with additional bits to form a CDI-40, EUI-48, or EUI-64. The IAB service was initiated in response to multiple concerns:

- We need to preserve OUI address space.
- The consumption rate of OUI was increasing.
- OUIs were being assigned to small companies that were not using them all.
- Customers desired another option, when less than 16 million addresses were needed.

The IAB reduces the consumption rate of OUIs, by effectively splitting a single OUI into 4096 useful subcomponents. A smaller fee is charged for the IAB, so as to encourage IAB usage and thus reduce the rate of OUI consumption.

However, the web pages are inconsistent due to:

a) Clerical error. The unanimously approved EUI-64 tutorial was not posted on the web site.

b) Errors have been found in the IAB assignment letter and tutorials.

While reviewing the needs for fixing (b), conflicting views on the intended usage of the IAB have arisen. This write-up expresses the usage conventions that minimize the number of required IEEE/RA services.

3. Documentation

The attached PDF files illustrates the contents of web pages proposed by this ballot. The IAB assignment letter documentation should also be changed as follows (xxx is the value assigned by the IEEE/RA):

OUI	Address block	Possible derivative identifiers:			
00-50-C2	xxx	CDI-40:	00-50-C2-xx-x0	to	00-50-C2-xx-xF
		EUI-48:	00-50-C2-xx-x0-00	to	00-50-C2-xx-xF-FF
		EUI-64:	00-50-C2-xx-x0-00-00-00	to	00-50-C2-xx-xF-FF-FF-FF

4. Rationale

Leveraging a single 36-bit IAB number assignment service across multiple environments (CDI-40, EUI-48, EUI-60, and EUI-64) has multiple benefits:

a) Fairness. The IAB can be used for MSC based standards (which typically are based on EUI-64) as well as LMSC standards (most of which are based on EUI-48).

b) Simple. The one IAB assignment service is sufficient to meet the diverse requirements of multiple unique number formats.

c) Clarity. A uniform approach towards OUI assignment reduction provides clarity to the users, when compared to the alternative of providing distinct {IAB-36, IAB-48, IAB-52} services for those using {EUI-48, EUI-60, and EUI-64} values respectively.

d) Practical. Smart users are likely to purchase the IAB-36, even though their intended usage is for EUI-60 or EUI-64. Its more valuable, so why not? Short of legal action (which is likely to fail), there is no way to prevent this usage.

Arguments have been raised, suggesting that the IEEE/RAC has priced the IAB too low, and should therefore discourage its usage. If this is indeed the case, usage of the IAB should be discontinued, not restricted. Hence, the second voting option.

5. Rebuttal

Others have proposed that IAB assignments should be in blocks of 4096 addresses, which has none of the preceding benefits. The argument is pricing equity, in that the users of an EUI-64 should not get “so many” addresses for such a low cost.

The preceding argument has several flaws:

a) Policy. The pricing model that justifies OUI fees is based on the cost of administering the assignments, not the value to the buyer. Any other policy is counter to the charter of the IEEE/RAC and likely to have serious administrative and legal consequences.

b) Precedence. Use of the OUI for EUI-64 and EUI-48 has already set a pricing precedent. In this case, the OUI is effectively 65536 times as valuable when used for EUI-64 assignments. This has never been raised as a concern.

c) Enforcement. Its not possible for the IEEE to micro-manage or enforce the usage of its identifiers. As such, there is no viable way to prohibit the use of an IAB within the EUI-64 address space.

The Institute of Electrical and Electronics Engineers, Inc., has been designated by the ISO Council to act as the registration authority for the implementation of International Standards ISO 8802.

September 26, 2005

Attn:

Following is the Individual Organizationally Unique Identifier (OUI) ~~and company_id~~ Address Block that you have requested from the IEEE. Please take a moment to verify that the information contained in the company addressed area (above) is correct. It is not entirely correct, please notify the IEEE Registration Authority immediately.

The user tutorials on our web site describe how the ~~OUI or company_id~~ can be used in conjunction with a number of standards and specifications. This assignment applies to all of these.

It should be stressed that the IEEE assignment committee has made every effort to ensure that there is no duplication of assignments, but does not guarantee that duplicate assignments have not occurred. We urge that a single, central administrative authority be established in your company to avoid duplication of identifiers based on your block of addresses.

The IEEE Registration Authority will assign an additional Individual ~~Organizationally Unique Identifier~~ to any organization requesting one, providing they submit a letter to the IEEE Standards Department stating that their company will not "ship" product in the new block assignment until well after they have reached (shipped) at least 95% of the original block assignment, in the context of a specified standard. Your company should ensure that large numbers of derived identifiers are not left unused.

Address Block is:

~~OUI/company id~~	~~Address Block~~
~~00-50-C2~~	~~???000-???FFF~~

The standard-specific tutorials describe the precise interpretation of the OUI ~~or company_id~~ values in context. For precise definition see the relevant standards.

The tutorials can be found on the web at:

http://standards.ieee.org/regauth/oui/tutorials

Sincerely,

Angela N. Weaver / IEEE Registration Authority

Request Form for an Individual Address Block of Addresses

To request an Individual Address Block, please submit request form 2. The Individual Address Block can be used in conjunction with a number of standards. It does not limit your right to use your assignment for both purposes. If there are no problems with the information on the application or the payment, the application will be processed within seven (7) working days.

As of 5 December 2003, companies requesting confidentiality for Individual Address Block assignments will be required to pay a fee for privacy. Assignees will also be required to pay an annual fee to maintain confidentiality.

Individual Address Blocks (IAB) assignment fee; $550.00 (US) plus $1,000.00 (US) to be listed as a private assignment, and $1,000.00 (US) annual fee for renewal of a private assignment.

Confidentiality Fee Pricing List			
Products	**Services**	**Fees**	**Total Due**
Individual Address Block (IAB)	Public Registry (company name and address on the public listing)	$550.00(US)	$550.00(US)
Individual Address Block (IAB)	Private Registry (company name and address not on the public listing)	$550.00(US) Plus $1,000.00(US)	$1,550.00(US)
IAB Annual Confidentiality Fee (Yearly)	Renewal* (for private registry only)	$1,000.00	$1,000.00

* Payment options for the renewal fee are credit card, check or wire transfer.

This single assignment applies to all of the following:

CDI-40: Context Dependent Identifier, 40-bits.
EUI-48: Extended Unique Identifier, 48-bits.
EUI-64: Extended Unique Identifier, 64-bits.

The IEEE Registration Authority will assign an additional IAB to any organization requesting one, providing they submit a letter on company letterhead to the IEEE Standards Department, stating that their company will not "ship" product in the new block assignment until well after they have reached (shipped) at least 95% of the block assignment, in the context of a specified standard. Your company should ensure that large numbers of derived identifiers are not left unused.

The IEEE does not accept requests for applicant-specified assignments. Please call the IEEE at the phone number listed if you have questions concerning the completion of this form.

Contact IEEE-SA
(IEEE Registration Authority)
URL: http://standards.ieee.org/regauth/oui/pilot-ind.html
(Modified: 11-March-2005)

Products & Services Home

Products
ShopIEEE
IEEE Standards Press

Services
Standards Online Subcriptions
Get IEEE 802®
Get IEEE/ANSI N42™
OUI, MAC or Ethernet and Other Registration

Reference Materials
Interpretations
Errata and Corrections
Downloadable Documents

IEEE Registration Authority Overview

Registration is the assignment of unambiguous names to objects in a way which makes the assignment available to interested parties.

Product	Description	Price	Processing Time
Organizationally Unique Identifier (OUI)/Company_id	An OUI / "company_id" is a 24-bit globally unique assigned number referenced by various standards. The OUI can be ~~is usually~~ concatenated with 24 or 40 bits by an Organization to create a ~~48-bit~~ number that is unique to a particular ~~piece of hardware~~ entity, forming an EUI-48 or an EUI-64. There are other uses of the OUI however, such as its use as a company identifier in the SNAP protocol. Click here (FAQs) and here (tutorials) for more information.	Public: $1,650 Private: $3,650*	7 business days from receipt of payment
Individual Address Block (IAB)	A 36-bit Individual Address Block is for ~~people who~~ organizations that need ~~less than 4097~~ no more than 2^{12} unique 48-bit numbers (EUI-48) or 2^{28} unique 64-bit numbers (EUI-64) and thus find it hard to justify buying their own OUI. The IAB ~~is a particular~~ consists of an OUI belonging to the IEEE Registration Authority, concatenated with 12 additional IEEE-provided bits, forming a unique 36-bit identifier. ~~leaving only 12 bits for the owner to assign to his (up to 4096) individual devices.~~ The owner can extend the IAB with 4, 12, or 28 bits, to form CDI-40, EUI-48, or EUI-64 identifiers respectively. Click here (FAQs) and here (tutorials) for more information.	Public: $550 Private: $1,550*	7 business days from receipt of payment
EtherType Field	The 16-bit Type Field provides a context for interpretation of the data field of an Ethernet/802.3 data frame (protocol identification). Refer to IEEE Std 802.3, Clause 3 and especially subclauses 3.1.1 and 3.2.6. See also IEEE Std 802 subclause 10.4. Click here for more information.	$2,500	90 Days
Standard Group MAC Address	A Universally Administered Address Block has been allocated for the assignment of Group MAC Addresses for use in Standards. This assignment is for standards developers only.Click here for more information.	No charge	7 business days
Logical Link Control (LLC)	Logical link control (LLC) protocol data units (PDUs) contain addressing information. This addressing information consists of two fields; the Destination Service Access Point (DSAP) address field, and the Source Service Access Point (SSAP) address field. Click here for more information.	No charge	150 Days
Manufacurer ID	Within a 64-bit section of the 1451.4 TEDS, called basic TEDS, the manufacturer of the transducer is defined with a 14-bit code called the manufacturer ID, along with manufacturer-assigned transducer model number, model letter, model version nember and serial number. Click here for more information.	$550	7 business days from receipt of payment
Universal Registration Number (URN)	The unique registration number is a 64-bit unique identifier contained in the memory devices, or nodes, in which IEEE 1451.4 TEDS data is stored. Because multiple nodes may be arrayed in a multi-drop network format, to allow memory capacity to be increased, or other functions to be added, the URN allows a number of nodes to be individually accessed by the system. Click here for more information.	$1,200	30 days from recipt of payment
IEEE Template	A template is a documented definition of the placement and significance of each piece of data stored within the TEDS memory. (see IEEE Std 1451.4.2004 subclause 5.3) The template is not contained within the TEDS data, but the TEDS data identifies which template is to be referenced in interpreting the TEDS data. Templates	$550	30 days from recipt of

	must be accessible to the program code, which is used to write and read the TEDS data, allowing that data to be properly packed for writing and unpacked subsequent to reading. Click here for more information.		payment
TDL Item	The TDL is a formal language similar to programming languages, but with considerably less looping and conditional control. This is because the entire purpose of the language is to map bits and not to implement general processing or mathematical capabilities. Click here for more information.	$550	30 days from recipt of payment

The IEEE Registration Authority formerly had administrative responsibility for the IEEE POSIX® Certification Authority.

To obtain background information on the IEEE Registration Authority, please click here.

The IEEE Registration Authority operates under the direction of the IEEE Standards Association Board of Governors. The IEEE Registration Authority Committee provides technical oversight for the IEEE Registration Authority Activities.

IEEE Registration Authority Activities

- Operating Procedures of the IEEE Registration Authority Committee (IEEE RAC)
- Subscribing to the Standards RAC Public Reflector - Mailing List Name: stds-rac-public (The public reflector can be used to voice opinions or ask questions related to industry-specific topics or standards within the Registration Authority group.)
- Archived Messages for Standards RAC Public Reflector
- IEEE Registration Authority Committee Private Pages (PASSCODE REQUIRED)

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 identifers:

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)

ACDE4823_16
```

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            | ext id          | field
|                           |                 |
| cdi[0] | cdi[1] | cdi[2] | cdi[3] | cdi[4] | order
|        |        |        |        |        |
|   AC   |   DE   |   48   |   23   |   45   | byte
```

Individual address block (IAB) based CDI-40 sequence

An individual address block (IAB) based 40-bit context dependent identifier (CDI-40) consists of two portions The 36-bit first portion of this value is specified by the IAB base 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 IAB base value is AC-DE-48-23-40-00 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
|                                       |    |
| cdi[0] | cdi[1] | cdi[2] | cdi[3] | cdi[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)
```

$ACDE482345_{16}$

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.

(IEEE Standards Systems/Network Staff)
URL: http://standards.ieee.org/regauth/oui/tutorials/EUI48.html
(Modified: 10-March-2005)

IEEE Navigational Bar

Guidelines for use of a 48-bit Extended Unique Identifier (EUI-48™)

General

The IEEE defined 48-bit extended unique identifier (EUI-48™) is a concatenation of a 24-bit Organizationally Unique Identifier (OUI) value administered by the IEEE Registration Authority and a 24-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 EUI-48 value can identify the manufacturer that provided the value h[1]. The IEEE/RAC has no control over the assignments of 24-bit extension identifiers and assumes no liability for assignments of duplicate EUI-48 identifiers assigned by manufacturers.

Application restrictions

Given the possibility of consuming all the EUI-48 identifiers, the IEEE/RAC places restrictions on their use. For new applications, EUI-48 identifiers are restricted to use in low volume applications, such as the identification of software interface standards or hardware model numbers.

While the number of EUI-48 identifiers is large, it is not inexhaustible, and an extended EUI-64 is available. Applications that use the EUI-48 identifier may require special review by the IEEE/RAC. See the *UseOfEUI* tutorial, *MAC-48 identifier restrictions*, for details.

The term EUI-48 is trademarked by the IEEE. Companies are allowed to use this term for commercial purposes, but only if their use of this term has been reviewed by the IEEE/RAC and the proposed products using the EUI-48 conform to these restrictions.

Manufacturer-assigned identifiers

The manufacturer identifier assignment allows the assignee to generate approximately 16 million unique EUI-48 values, by varying the last 24 bits. The IEEE intends not to assign another OUI value to a manufacturer of EUI-48 values until the manufacturer has consumed, in product, the preponderance (more than 90%) of this block. It is incumbent upon the manufacturer to ensure that large portions of the block are not left unused in manufacturing.

48-BIT EXTENDED UNIQUE IDENTIFIER FORMAT TUTORIAL

General

The IEEE defined 48-bit global identifier (EUI-48) is assigned by a manufacturer who has been assigned an OUI value by the IEEE Registration Authority. The 48-bit identifier is a concatenation of a 24-bit OUI value assigned by the IEEE Registration Authority and a 24 bit extension identifier assigned by the organization with that OUI assignment.

The IEEE administers the assignments of 24-bit OUI values. The assignments of these values are public, so that a user of a EUI-48 value can identify the manufacturer that provided the value. The IEEE has no control over the assignments of 24-bit extension identifiers and assumes no liability for assignments of duplicate EUI-48 identifiers.

Distribution restrictions

The IEEE has no established policies on the redistribution of EUI-48 values or range of values through third parties.

OUI based identifier sequence

An OUI based 48-bit global identifier (EUI-48) 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 24-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 23-45-67. The EUI-48 value generated from these two numbers is AC-DE-48-23-45-67.

Some standards specify an EUI-48 by a string of six bytes, labeled here as *eui*[0] through *eui*[5]. For those standards, the format of the EUI-48 is illustrated below. Although different standards may specify different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
Value: AC-DE-48-23-45-67

|            OUI            |   extension identifier    | field
|                           |                           |
| eui[0] | eui[1] | eui[2] | eui[3] | eui[4] | eui[5] | order
|        |        |        |        |        |        |
|   AC   |   DE   |   48   |   23   |   45   |   67   | byte
```

IAB based identifiers

An individual address block (IAB) based 48-bit global identifier (EUI-48) consists of two portions. The 36-bit first portion of this value is specified by the IAB base value assigned to the manufacturer by the IEEE Registration Authority; the 12-bit second portion of this identifier is assigned by the manufacturer.

For example, assume that a manufacturer's IEEE-assigned IAB base value is AC-DE-48-23-40-00 and the manufacturer-selected extension identifier for a given component is 5-67. The EUI-48 value generated from these two numbers is AC-DE-48-23-45-67.

Some standards specify an EUI-48 by a string of six bytes, labeled here as *eui*[0] through *eui*[5]. For those standards, the format of the EUI-48 is illustrated below. Although different standards may specify different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
IAB:       AC-DE-48-23-40-00(smallest) through
           AC-DE-48-23-4F-FF(largest)
extension: 5-67
EUI-48:    AC-DE-48-23-45-67
```

```
Value: AC-DE-48-23-45-67

  |                  IAB                  |  extension   | field
  |                                       |              |
  | eui[0] | eui[1] | eui[2] | eui[3] | eui[4] | eui[5] | order
  |        |        |        |        |        |        |
  |   AC   |   DE   |   48   |   23   |   45   |   67   | byte
```

Numerical formats

Other standards specify an EUI-48 to be a numerical value, upon which computations (such as base/bounds or bit selections) can be performed. For those standards, the format of the EUI-48 is illustrated below:

```
  Value: ACDE48234567(16)

  |            OUI            |   extension identifier   | field
  |                           |                          |
  |   AC   |   DE   |   48    |   23   |   45   |   67   | hex
```

---

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

(IEEE Standards Systems/Network Staff)
URL: http://standards.ieee.org/regauth/oui/tutorials/EUI48.html
(Modified: 10-March-2005)

Guidelines for use of a 64-bit Extended Unique Identifier (EUI-64™)

General

The IEEE defined 64-bit extended unique identifier (EUI-64) is a concatenation of a 24-bit Organizationally Unique Identifier (OUI) value administered by the IEEE Registration Authority and a 40-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 EUI-64™ value can identify the manufacturer that provided the value[1]. The IEEE has no control over the assignments of 40-bit extension identifiers and assumes no liability for assignments of duplicate EUI-64 identifiers assigned by manufacturers.

Application restrictions

Given the minimal probability of consuming all the EUI-64 identifiers, the IEEE/RAC places minimal restrictions on their use within standards. Their major use is to distinctively identify hardware instances of devices.

They may also be used for other purposes where a unique identifier is desired. However, they may not be used to identify autonomously generated objects such as:

1. Transaction identifiers or other ephemeral identifiers.
2. Sequence numbers that distinctively identify power or reboot cycle instances.

However, EUI-64 identifiers could be concatenated with a locally managed 64-bit identifier to create distinct 128-bit (or larger) unique object or file identifiers. Such use is not discouraged.

The IEEE/RAC reviews draft IEEE standards for correctness and clarity of EUI-64 usage. The IEEE/RAC does not restrict the use of EUI-64 identifiers within standards, if the standard conforms to the aforementioned restrictions. If the EUI-64 is referenced within non-IEEE standards, there shall not be any reference to IEEE unless approved by the IEEE/RAC. The main purpose of IEEE/RAC review is to ensure sufficiently efficient usage of assigned address spaces.

Usage restrictions

The EUI-64 is a trademarked IEEE term and there are a number of restrictions on EUI-64 usage; see <<application form>> for details. These restrictions are subject to change on a per-purchase basis.

Application documentation

As a condition for receiving an OUI assignment, a manufacturer of EUI-64 values accepts the following responsibilities:

1. This documentation must be readily available (at no cost) to any purchaser of EUI-64 values.
2. The manufacturer's part specification should include an unambiguous description of how the EUI-64 value is accessed (pin and/or address descriptions).

Manufacturer-assigned identifiers

The manufacturer identifier assignment allows the assignee to generate approximately 1 trillion (10^{12}) unique EUI-64 values, by varying the last 40 bits. The IEEE intends not to assign another OUI value to a manufacturer of EUI-64 values until the manufacturer has consumed, in product, the preponderance (more than 90%) of this block. It is incumbent upon the manufacturer to ensure that large portions of the block are not left unused in the consumption process.

64-BIT EXTENDED UNIQUE IDENTIFIER FORMAT TUTORIAL

OUI based identifier sequence

An OUI based 64-bit global identifier (EUI-64) 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 40-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 23-45-67-AB-CD. The EUI-64 value generated from these two numbers is AC-DE-48-23-45-67-AB-CD.

Some standards specify an EUI-64 by a string of eight bytes, labeled here as *eui*[0] through *eui*[7]. For those standards, the format of the EUI-64 is illustrated below. Although different standards may specify different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
OUI:       AC-DE-48
extension: 23-45-67-AB-CD
EUI-64:    AC-DE-48-23-45-67-AB-CD
```

OUI			extension identifier					field
eui[0]	eui[1]	eui[2]	eui[3]	eui[4]	eui[5]	eui[6]	eui[7]	order
AC	DE	48	23	45	67	AB	CD	hex

Individual address block (IAB) based identifiers

An individual address block (IAB) based 64-bit global identifier (EUI-64) consists of two portions. The 36-bit first portion of this value is specified by the IAB value assigned to the manufacturer by the IEEE Registration Authority. The 28-bit second portion of this identifier is assigned by the manufacturer.

For example, assume that a manufacturer's IEEE-assigned IAB base value is AC-DE-48-23-40-00 and the manufacturer-selected extension identifier for a given component is 5-67-AB-CD. The EUI-64 value generated from these two numbers is AC-DE-48-23-45-67-AB-CD.

Some standards specify an EUI-64 by a string of eight bytes, labeled here as *eui*[0] through *eui*[7]. For those standards, the format of the EUI-64 is illustrated below. Although different standards may specify

different bit-transmission orders, bytes are normally transmitted in an ascending index-value order.

```
IAB:       AC-DE-48-23-40-00(smallest) through
           AC-DE-48-23-4F-FF(largest)
extension: 5-67-AB-CD
EUI-64:    AC-DE-48-23-45-67-AB-CD
```

IAB				extension identifier				field
eui[0]	eui[1]	eui[2]	eui[3]	eui[4]	eui[5]	eui[6]	eui[7]	order
AC	DE	48	23	45	67	AB	CD	hex

Numerical formats

Other standards specify an EUI-64 to be a numerical value, upon which computations (such as base/bounds or bit selections) can be performed. For those standards, the format of the EUI-64 is illustrated below:

Value: $ACDE48234567ABCD_{16}$

OUI			extension identifier				field
AC	DE	48	23	45	67	AB	CD	hex

Restricted encapsulated values

To support encapsulation of EUI-48™ and MAC-48 values within small subsets of the EUI-64 values, the first two bytes of the manufacturer's extension identifier must not be FF-FF or FF-FE. Thus, the 64-bit values of the following form are never assigned EUI-64 values:

XX-XX-XX-FF-FE-YY-YY-YY (an EUI-48 extension)

XX-XX-XX-FF-FF-YY-YY-YY (a MAC-48 extension, obsolete)

The letters 'X' and 'Y' represent hexadecimal digits and show how the EUI-48 value can be unambiguously encapsulated within the EUI-64 value; the 'X' and 'Y' digits represent the OUI and extension-identifier portions of the EUI-48/MAC-48 values, respectively.

This allows EUI-48 identifiers to be encapsulated and transported as (otherwise unassigned) EUI-64 identifiers. The intent is to enable migration to a single form of OUI based globally unique 64-bit identifiers.

Encapsulated EUI-48 values

An EUI-48 identifier can be encapsulated and placed within an EUI-64 container. For example, assume that a manufacturer's IEEE-assigned OUI value is AC-DE-48 and the manufacturer-selected extension identifier for a given item is 23-45-67. The EUI-48 value generated from these two numbers is AC-DE-48-23-45-67, whose byte representationn is illustrated below:

```
EUI-48 value: AC-DE-48-23-45-67


|            OUI           |   extension identifier   | field
|                          |                          |
|   AC   |   DE   |   48   |   23   |   45   |   67   | hex
```

An EUI-64 value is generated by concatenating OUI, FF-FE, and extension-identifier values. The byte representation for this EUI-64 value is illustrated below:

```
Encapsulating EUI-64 value: AC-DE-48-FF-FE-23-45-67


|            OUI           |    EUI label    |   extension identifier   | field
|                          |                 |                          |
|   AC   |   DE   |   48   |   FF   |   FE   |   23   |   45   |   67   | hex
```

Encapsulated MAC-48 values (obsolete)

A MAC-48 identifier can be encapsulated and placed within an EUI-64 container. For example, assume that a manufacturer's IEEE-assigned OUI value is AC-DE-48 and the manufacturer-selected extension identifier for a given item is 23-45-67. The EUI-48 value generated from these two numbers is AC-DE-48-23-45-67, whose byte representation is illustrated below:

```
EUI-48 value: AC-DE-48-23-45-67


|            OUI           |   extension identifier   | field
|                          |                          |
|   AC   |   DE   |   48   |   23   |   45   |   67   | hex
```

An EUI-64 value is generated by concatenating OUI, FF-FF, and extension-identifier values. The byte representation for this EUI-64 value is illustrated below:

```
Encapsulating EUI-64 value: AC-DE-48-FF-FF-23-45-67


|            OUI           |    EUI label    |   extension identifier   | field
|                          |                 |                          |
|   AC   |   DE   |   48   |   FF   |   FF   |   23   |   45   |   67   | hex
```

Encapsulated EUI-60 values

An EUI-60 identifier can be encapsulated and placed within an EUI-64 container. For example, assume that a manufacturer's IEEE-assigned OUI value is AC-DE-48 and the manufacturer-selected extension identifier for a given item is 2.3.4.5..6.7.A.B.C. The EUI-60 value generated from these two numbers is AC-DE-48:2.3.4.5.6.7.A.B.C, whose nibble representation is illustrated below:

```
EUI-60 value: AC-DE-48:2.3.4.5.6.7.A.B.C


|                OUI           |             extension identifier           | field
|                              |                                            |
|  0 |  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 | 11 | 12 | 13 | 14 | nibbl
```

```
    A    C    D    E    4    8    2    3    4    5    6    7    A    B    C | hex
```

An encapsulating EUI-64 value is generated by appending the EUI-60 with a 4-bit zero value, as illustrated below:

```
Encapsulating EUI-64 value: AC-DE-48-23-45-67-AB-C0


|            OUI           |           extension identifier   | field          |
|                          |                                  |                |
|   AC   |   DE   |   48   |   23   |   45   |   67   |   AB   |   C0   | hex
```

Unassigned EUI-64 values

The all-zeros and all-ones EUI-64 values will never be assigned and may be used to represent NULL EUI-64 address values. As an example, either of these values could represent an uninitialized value, before that location is accessible or properly initialized by hardware/firmware. Similarly, either value could represent invalid, when an optional EUI-64 value is not supplied.

---

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

(IEEE Standards Systems/Network Staff)
URL: http://standards.ieee.org/regauth/oui/tutorials/EUI64.html
(Modified: 10-March-2005)