Extract from guid paper 20210414.

# GUID encoding requirements and characteristics

# ISO/IEC 9834-8:2014 Annex B

Annex B provides functional requirements for name-based versions of a [UUID](#_bookmark265).

* 1. The name-based [UUID](#_bookmark265) is meant for generating a [UUID](#_bookmark265) from a name that is drawn from, and unique within, some name space. The concept of name and name space should be broadly construed, and not limited to textual names. The mechanisms or conventions for allocating names from, and ensuring their uniqueness within, their name spaces are beyond the scope of this Specifica- tion. NOTE – In order to avoid recursion problems, name-based [UUIDs](#_bookmark265) should not be generated from an [OID](#_bookmark259) that ends with a [UUID](#_bookmark265) which is name-based.
  2. The properties of name-based [UUIDs](#_bookmark265) generated in accordance with clause 14 and with a suitably chosen namespace will be as follows:
     + The [UUIDs](#_bookmark265) generated at different times from the same name in the same namespace will be equal;
     + The [UUIDs](#_bookmark265) generated from two different names in the same namespace will be different with very high probability;
     + The [UUIDs](#_bookmark265) generated from the same name in two different namespaces will be different with very high probability;
     + If two name-based [UUIDs](#_bookmark265) are equal, then they were generated from the same name in the same namespace with very high probability.

# RFC 4122 clause 4.3

RFC [4122[16](#_bookmark215)] provides an algorithm to create a name-based version. ISO/IEC 9834- 8:2014[[17](#_bookmark216)] specifies the requirements for the encoding and interpretation of a [UUID](#_bookmark265). There are two versions of [UUIDs](#_bookmark265) considered: random-number-based and name-based. Both ver- sions meet the functional requirement of uniqueness.

# UUID versions and critical characteristics

[UUID](#_bookmark265) versions available and brief description.

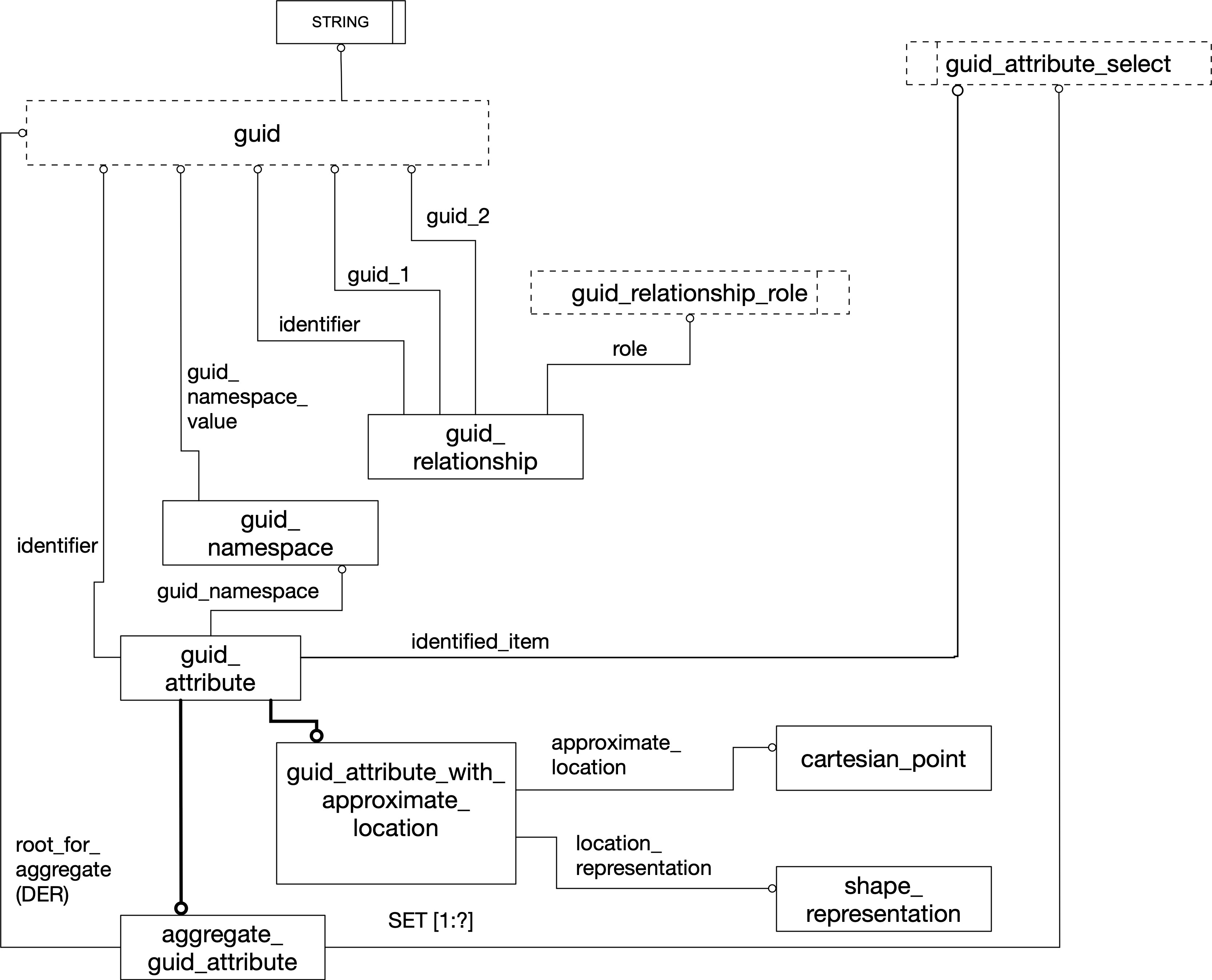
* + - 1. (date-time - MAC address)
      2. (DCE Security)
      3. (MD5 hash - namespace)
      4. (random) currently used by ST and MAC
      5. (SHA-1 hash - namespace) Version 5 produces the same result each time the same name in the same namespace is transformed. The namespace identifier is itself a UUID. The specification provides UUIDs to represent the namespaces for URLs, fully qualified domain names, object identifiers, and X.500 distinguished names; but any desired UUID may be used as a namespace designator.

# Proposed STEP information model summary

# Recommended encoding structure

Version 5 uses two arguments:

* + - * Domain key
      * Name string



**Fig. 4.** The key proposed entities and types rendered in [EXPRESS](#_bookmark246)-G.

A [GUID](#_bookmark248) value is dependent on the combination of Domain key and Name string. The Domain key is exchanged in the data set and is proposed to be a [GUID](#_bookmark248) based on a private string maintained by the software application owner and generated using the version 5 methodology. The Domain key is only dependent on the sourcing application[11](#_bookmark103). Neither the domain key nor the name string is required to be exchanged in the data set. The name string is proposed to be the internal [OID](#_bookmark259) or unambiguous path to the internal [OID](#_bookmark259) from the intrnal root object in the [CAD](#_bookmark239) model. The [GUID](#_bookmark248) proposed herein is a 36 character string which supports the requirements of version 5 encoding specified in [[16](#_bookmark215)], [[17](#_bookmark216)], and [[8](#_bookmark207)].

11CAD applications have migrated to proprietary, locked, internal instance databases; in the majority of use cases, there is no third party translator with access to internal [CAD](#_bookmark239) [OID](#_bookmark259)s.

# Proposed new entities for the STEP information model

Figure [4](#_bookmark101) is an illustration of the core entities and types in the proposed model.

# guid namespace

The guid namespace ENTITY provides the capability to store and exchange a namespace [GUID](#_bookmark248) for the application. An informal proposition will be provided to require implemen- tation conformance.

The mapping from an object in an internal application software model to an instance in the STEP product model structure is application dependent. The domain key is internal to the application software model. Therefore, there is no attempt to model any particular structure in the STEP product model as being an input to the algorithm that calculates the value of guid attribute.attribute value nor of aggregate guid attribute.attribute value.

# guid attribute

The guid attribute ENTITY provides an assignment of the tuple ([GUID](#_bookmark248), namespace) to a target instance. The guid attribute identifier attribute complies with[[17](#_bookmark216)] and [[16](#_bookmark215)]. An informal proposition will be provided to require implementation conformance.

# aggregate guid attribute

The aggregate guid attribute provides an assignment of the tuple ([GUID](#_bookmark248), namespace) to a collection of target instances[12](#_bookmark108). The critical or root node the collection is dependent upon in the product data set is specified as the root for aggregate. This entity is under review as its use case is unclear at this writing. The application of aggregate id attribute in the topology domain could have been implemented by an instance of connected face - sub set (citation) but there are other more critical use cases e.g., a collection of all the data associated with a shape aspect or a collection of related characteristics[13](#_bookmark109). The use of the generic term aggregate should be replaced with a more appropriate term

# guid relationship

The guid relationship ENTITY provides the capability to build a graph[14](#_bookmark111) of [GUID](#_bookmark248)s where the guid relationship is conceptually an edge and [GUID](#_bookmark248)s are conceptually vertices[15](#_bookmark112). A role attribute is provided as an enumeration for centralized management in the ISO TC 184/SC 4 WG 12 development process. The proposed values are derive from, merge, split, and supersedes. A [GUID](#_bookmark248) is provided as an identifier so that the guid relationship records may be referenced in a graph of [GUID](#_bookmark248)s.

**Example 10.** A member of guid relationship with a role of ”supersedes” is provided to indicate that an instance replaces another instance.

14The graph is not explicitly represented.

15A more formal graph-theoretic basis is required for publication

**Example 11.** Two instances of guid relationship, each with the role of ”merge” and each specifying the same target [GUID](#_bookmark248), are provided to indicate that two instances may be merged into one instance.[16](#_bookmark115)

**Example 12.** Two instances of guid relationship, each with the role of ”split” and each specifying the same source [GUID](#_bookmark248) are provided to indicate that one instance has been split into two instances[17](#_bookmark116).

# Proposed application behavior

# Namespace management

Each application that creates STEP data sets must create a namespace [GUID](#_bookmark248) for their appli- cation in the context of the using organization. The details of that transaction are outside thescope of this document. The only requirement proposed for namespace in this document is that the namespace be present in a data set and encoded as discussed in [10.2.1](#_bookmark105).

**Example 16.** This example uses only the Application namespace and does not include clear-text encoding of the application meta-data in the file. None of the instances related to the datum are included.(newlines added for presentation, white space is ignored)[22](#_bookmark130).

Filename: "the cad FILENAME";

Type: "datum", Cad object ID: "ID 12345";

#22=GUID\_NAMESPACE(9e624f87-66c3-5064-aff9-dea6c4ab9946); #33=DATUM(...);

#44=GUID\_ATTRIBUTE(68efd509-fe3c-57ac-87b9-204763d5e4bf, #33,#22,$,$);

# Object management

Some flexibility is supported with this proposed model by providing the ability to assign a [GUID](#_bookmark248) to a single object or to a collection of objects. The members of the collection of ob- jects must be connected.

Each organization should assume that if they don’t own the object, they shall not modify the object. They can obviously propose changes to the owner.

22spaces converted to dashes

# GUID reuse prohibited

A [GUID](#_bookmark248) represents the complete path to the [CAD](#_bookmark239) object from the design root with a unique key assigned for that path. When an object is deleted that path is deleted and the internal object index in the authoring software will declare a ’deleted’ value for that key so as to prevent re-use.

# GUID management in context of data set revisions

We propose that a revision to a design shall regenerate [GUID](#_bookmark248)s that are identical to the previous design where the design object is not impacted by the revision.[24](#_bookmark135) [GUID](#_bookmark248)s can be unambiguously mapped to the existing STEP identification rather than to the internal [CAD](#_bookmark239) object ids where appropriate.

24Model structures that are networks present challenges.

# Internal pre-processor export rules

These rules are in addition to the rules inherited from the [GUID](#_bookmark248) management in context of data set revisions above. Annotation cannot be merged or split during translation in order to preserve the consistency of the associated UUID. Any individual geometry instance that is tagged with a UUID must be preserved during translation. In the case that the application is using a shape aspect instance as a proxy for the original element, the shape aspect, item identified representation usage, topological item instance, and the geometrical item instance must be preserved. Any aggregate geometry instance that is tagged with a UUID must be preserved during translation. In the case that the application is using a shape aspect instance as a proxy for the original aggregate, the shape aspect, item identified representation usage and the geometry instance must be preserved. The application must export any preserved [GUID](#_bookmark248) in the model. The application must export any created [GUID](#_bookmark248) in the model. The decision of what [CAD](#_bookmark239) objects to assign [GUID](#_bookmark248)s to is implementation dependent.

# External pre-processor export rules

The application must export any preserved [GUID](#_bookmark248) in the native [CAD](#_bookmark239) model. The application must export any [GUID](#_bookmark248) assignment of [CAD](#_bookmark239) objects specified for the translator configura- tion.

**Example 22.** A configuration file may specify that all datums shall have a [GUID](#_bookmark248) assigned.

# The post-processor import rules

The application may create a [GUID](#_bookmark248) if it does not exist on import[25](#_bookmark142). The application must preserve incoming [GUID](#_bookmark248)s. The applications that support change request shall support design change management as specified in AP 242[26](#_bookmark143).

# STEP data set state change by an application

The application that changes the state of a STEP data set respects the [GUID](#_bookmark248) population from the transmitting organization, decides what new objects need [GUID](#_bookmark248)s and puts [GUID](#_bookmark248)s only on those objects when it generates an updated STEP data set.

# Downstream applications that create data

Downstream applications that create data shall conform to version 5 namespace and name string requirements for [GUID](#_bookmark248) creation.

25There may be a configuration file to specify the types for which [GUID](#_bookmark248) shall be created.

26It may be necessary to pre-process the measurement results from manufacturing and from manufacturing process planning to generate the correct change request records.

# STEP data set state change requests

The applications that request changes to a STEP data set shall support design change man- agement as specified in AP 242. It may be necessary to pre-process the measurement re- sults from manufacturing and from manufacturing process planning to generate the correct change request records.

# New declarations for the STEP resource model

# This guid\_attribute\_select list is subject to serious debate.

TYPE guid\_attribute\_select = EXTENSIBLE GENERIC\_ENTITY SELECT (application\_context,

characterized\_object, characterized\_object\_relationship, context\_dependent\_shape\_representation, datum\_feature,

datum\_system, derived\_unit,

dimension\_related\_tolerance\_zone\_element, dimensional\_characteristic\_representation, dimensional\_location,

dimensional\_size, founded\_item, general\_property, general\_property\_relationship, geometric\_tolerance,

geometric\_tolerance\_auxiliary\_classification, geometric\_tolerance\_relationship, gps\_filter,

gps\_filtration\_specification, invisibility, item\_identified\_representation\_usage, limits\_and\_fits, maths\_value\_with\_unit, measure\_qualification, measure\_with\_unit,

named\_unit, placed\_datum\_target\_feature, plus\_minus\_tolerance, product, product\_definition,

product\_definition\_formation, product\_definition\_formation\_relationship, product\_definition\_relationship, property\_definition, property\_definition\_representation,

representation, representation\_context, representation\_item, representation\_relationship, runout\_zone\_orientation, shape\_aspect, shape\_aspect\_relationship, tolerance\_value,

tolerance\_with\_statistical\_distribution, tolerance\_zone\_definition, tolerance\_zone\_form, topological\_representation\_item);

END\_TYPE;

Note: geometric representation item is intentionally omitted as for engineering applica- tions, the shape aspect provides identification for members of geometric representation - item.

TYPE guid\_relationship\_role = ENUMERATION OF (supersedes,

merge,

same\_as,--to be added soon

split, derive\_from);

END\_TYPE;

ENTITY aggregate\_guid\_attribute SUBTYPE OF (guid\_attribute);

aggregate\_identified\_item : SET [1 : ?] OF guid\_attribute\_select; DERIVE

root\_for\_aggregate : guid := SELF\guid\_attribute.identifier; END\_ENTITY;

ENTITY guid\_attribute; identifier : guid;

identified\_item : guid\_attribute\_select; guid\_namespace : guid\_namespace;

END\_ENTITY;

ENTITY guid\_attribute\_with\_approximate\_location SUBTYPE OF(guid\_attribute); approximate\_location : cartesian\_point; location\_representation : shape\_representation;

WHERE

WR1: location\_representation in using\_representations(approximate\_location); END\_ENTITY;

ENTITY guid\_namespace; guid\_namespace\_value : guid;

UNIQUE

UR1 : guid\_namespace\_value; END\_ENTITY;

TYPE guid = STRING(36) FIXED; END\_TYPE;

ENTITY guid\_relationship; identifier : guid; guid\_1 : guid;

guid\_2 : guid;

role : guid\_relationship\_role; UNIQUE

UR1 : identifier;

WHERE

IP1 : acyclic\_guid\_relationship(SELF, guid\_1, guid\_2);

-- A guid\_relationship shall be acyclic END\_ENTITY;

**Universally Unique Identifier** a Universally Unique Identifier is an identifier that is unique across both space and time [IETF RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace].. [9](#_bookmark48), [41](#_bookmark223)