Automatic Generation of Unique Device IDs

A Research Report

- Abhijit A. Bokil

Introduction

This brief Research Report discusses the following

- What is a unique ID?
- Advantages of UIDs
- UID standards
- Issues
- Tradeoffs
- Custom approaches
- Summary

What is a unique ID?

In the world of computing systems, it is often necessary to assign an identification number to a piece of software data or a hardware device/component. For instance, a GPS vehicle tracking system might store location data in a database by assigning a timestamp to individual pieces of location data. The manufacturer of a certain hardware component may assign a randomly generated numerical ID to every unit of the component that emerges from its assembly line. Another popular illustration of a unique ID is India's Aadhaar number, which assigns a unique ID to each registered citizen.

Advantages of UIDs

Since UIDs are essentially names, assigning each item of a certain type with a unique ID allows that item to be uniquely tracked from creation to obsolescence. When information about items with

unique IDs is stored in a database, the UID can be used as a primary key; it is possible that for two separate entries, every other parameter or characteristic is identical, excepting the UID, allowing the user to distinguish between the 2 entries at all times.

In the hardware domain, UIDs are employed for the following reasons:

- Identification
  - For instance, the IMEI number assigned to every mobile phone contains fields with origin and model number so that every mobile can be uniquely identified
- Traceability
  - A stolen mobile phone can be traced/tracked via its IMEI
  - Over-the-air (OTA) firmware/software updates can be driven based on unit IDs (e.g Set top boxes)
- Manufacturing process tracking
  - If there are multiple PCB variants of a device, the specific PCB type used in a unit can be determined from its unique ID if the ID contains that PCB type number. This feature is particularly useful in the event of a device recall by the manufacturer.

UID Standards

In many domains, UIDs are directly assigned by governing technical bodies, and such groups of UIDs may be purchased in bulk by device vendors.

There are traditional ways of generating a unique ID, such as a counter that increases at each call or a timestamp pulled from an operating system. These approaches are in use today and might serve limited purposes. Associated shortcomings are two units being assigned the same timestamp on different production lines or counters having constrained ranges.

UIDs may be arbitrarily assigned by a device manufacturer, or they may be globally unique. In the former instance, two devices from two different vendors may have the same ID, leading to confusion.

Thus UIDs are generally unique at a global level, i.e. all vendors of a certain device type agree on a identification/numbering scheme for their particular industry/domain. Typically, a standards body with vendors as members defines and delineates an ID standard which is then adopted by all certified members.

UUIDs are Universally Unique IDs (or Globally Unique IDs/GUIDs) used widely in the hardware/software domain. These are 128-bit (32 hex digit) labels defined by RFC4122, which is the relevant IETF standard. (https://www.rfc-editor.org/rfc/rfc4122)

Such UUIDs are often represented in the 8-4-4-4-12 hex format.

(e.g. ***ab347cc5-36f4-fe34-00a3-77dd43ea2110***)

RFC4122 guarantees uniqueness across space and time. The standard also pays attention to factors such as high allocation rates of the algorithm and ease of programming.

Version 1

(DateTime + MAC) Here the UID consists of a combination of the timestamp (e.g. milliseconds elapsed since the beginning of an agreed upon epoch) and the MAC address of

the device, the latter being organizationally unique and assigned by IEEE to vendors. RFC4122 allows the vendor ID to be replaced by a random number

Version 3 & 5

These UUID versions involve hashing, a method that takes as input a string or number, and generates a fixed-size value (ID) as output. Version 3 uses the MD5 algorithm, while version 5 uses SHA-1. There is no random component and for the same inputs, these UUID versions always return the same IDs.

Version 4

This type of UUID is generated using a random/pseudo-random number generator. It is the most commonly used UUID version. In the 128-bit v4 UUID, 6 bits are fixed by RFC4122 for variant/version, implying that the probability of ID collision is $1/2^{122}$ which is a truly insignificant number.

Issues

Uniqueness

It is imperative that every ID generated by the UID generation algorithm/system be absolutely unique. However, if the deterministic process/algorithm by which the ID was generated is well-known, then it leads to a condition wherein anyone with knowledge of the process can replicate the ID, perhaps for nefarious purposes. Concomitantly, there is a loss of anonymity as the structure of the ID is known.

Randomness

This is a measure of the likelihood that two IDs generated by the same algorithm will not coincide/match. In order to implement randomness in UIDs, systems may employ truly-random or pseudo-random number generators.

Scalability

In a world where there are already millions of connected devices, it becomes necessary to assign UIDs speedily and reliably. This means multiple test/assembly lines assigning UIDs simultaneously via the same algorithm.

Tradeoffs

Uniqueness and randomness cannot both be achieved at once by an assignment algorithm. Either a UID is fully deterministically allocated making it unique and allowing the device to be traceable/trackable, etc. or the UID is fully randomly allocated, with a miniscule probability of UID collision, but also implying that there is no discernable structure.

In practice, UIDs are a combination of a unique component and a randomly assigned component.

For instance, a 128-bit device UID may consist of 40 bits of uniqueness & 88 bits of randomness, making for the format illustrated below.

UNIQUE				RANDOM
VENDOR_ID 10 bits	MODEL_ID 4 bits	MANUF_DATE 24	PCB_VERS. 2 bits	<8d6559ff3e4b283e746fe0> 88 bits

Other issues:

When multiple ID servers are used to assign IDs to devices rolling off the assembly lines, the line ID must naturally form a part of the UID. Additionally, if the UID contains a timestamp, the time must be fetched either from an onboard RTC or from the test-host system itself. This time must in turn be synchronized with UTC using time protocols like NTP. Also, all test-lines must be synchronized.
For devices which are manufactured in significant quantities, the probability of collision must be assessed before employing any particular randomization method.

Custom approaches

While RFC4122 clearly outlines the structure of a UUID, other more customised approaches are also prevalent. The 'ordered' or 'combined-time' UUID allows for a timestamp (32-40 bits) followed by a randomized part (96-88 bits). The primary advantage of beginning a UUID with a timestamp is that when placed in a database, the UUIDs can be chronologically sorted; such UUIDs can also be stored more efficiently in indexed DB columns than completely random IDs.

There is no commonly agreed upon specification for such custom approaches.

Summary

UIDs are used to uniquely and reliably identify individual devices. There are several formally specified versions of UUIDs as per the RFC4122 standard. Timestamping allows a vendor to uniquely identify every device, and this is often used in combination with a randomly generated ID part, which significantly reduces the probability of ID collision. There is always a tradeoff between uniqueness and randomness, and heuristic considerations will often govern the approach a vendor will take to UID assignment. Apart from the standard methods, a vendor/service provider may choose a custom algorithm more suited to specific needs.

References

- https://www.sohamkamani.com/uuid-versions-explained/
- https://www.uuidtools.com/uuid-versions-explained
- https://muradiodev.medium.com/unraveling-the-intricacies-of-uuid-the-art-of-creating-uniqueness-in-chaos-6173391cea45
- https://www.rfc-editor.org/rfc/rfc4122
- https://en.wikipedia.org/wiki/Universally_unique_identifier