

# UUID Versions Explained

## What are the different UUID versions and how to use them?

There are 5 UUID version plus one common unofficial version. We will explain each of these versions as well as their purpose and when to use them. For more details, see the official [UUID RFC 4122](#).

### UUID Version-1

**Version-1 is based on the current time and the MAC address for the computer or "node" generating the UUID.**

RFC 4122 states timestamp is number of nanoseconds since October 15, 1582 at midnight UTC. Most computers do not have a clock that ticks fast enough to measure time in nanoseconds. Instead, a random number is often used to fill in timestamp digits beyond the computer's measurement accuracy. When multiple version-1 UUIDs are generated in a single API call the random portion may be incremented rather than regenerated for each UUID. This ensures uniqueness and is faster to generate.

The last 12 hex digits of a UUID string represent the **MAC address** of the node. In some implementations (including the [UUID generator](#) on this site) a random MAC address is used instead of the node's actual MAC.






MAC Address and creation time can be extracted using our [UUID decode tool](#). MAC Addresses are unique to the computer generating them. By including a MAC address in the UUID you can be sure that two different computers will never generate the same UUID. Because MAC addresses are globally unique, version-1 UUIDs can be traced back to the computer that generated them.


#### Table of Contents

- [Version-1 UUID](#)
- [Version-2 UUID](#)
- [Version-3 vs Verion-5 UUID](#)
- [Version-4 UUID](#)
- [Timestamp-first UUID](#)
- [Nil UUID](#)

### UUID Version-1 Record Layout

**123e4567-e89b-12d3-a456-426655440000**  
**xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx**

-  Low Time
-  Mid Time
-  High time and version
-  Clock sequence and variant
-  Node

 UUIDTools.com

Records Layout for UUID version-1.

### UUID Version-2

Version-2 is called "DCE security" UUIDs in RFC 4122. Further details are not provided in the official RFC so our implementation omits version-2. [Version-2 specifications](#) are, however, published by DCE.

### UUID Version-3 vs UUID Version-5

Version-3 and version-5 are generated based on a "namespace" and unique "name". **Namespace and name are concatenated and hashed. There is no temporal or random component to either versions so the same input produces the same output every time.**

- namespace — a UUID
- name — can be anything

**What is the difference between version-3 and version-5?**

- Version-3 UUIDs are based on an [MD5](#) hash of the name and namespace.

- Version-5 UUIDs are based on a [SHA-1](#) hash of the name and namespace.

A SHA-1 hash is too long to be used in a UUID so it is truncated.

The UUID specification establishes 4 pre-defined namespaces. The pre-defined namespaces are:

- DNS — 6ba7b810-9dad-11d1-80b4-00c04fd430c8
- URL — 6ba7b811-9dad-11d1-80b4-00c04fd430c8
- OID — 6ba7b812-9dad-11d1-80b4-00c04fd430c8
- X.500 DN — 6ba7b814-9dad-11d1-80b4-00c04fd430c8

When a pre-defined namespace identifier is used we convert that identifier to a UUID internally.

When using our [version-3](#) or [version-5](#) UUID generators you will be asked to enter a "namespace" (or pre-defined UUID identifier) and "name".

## UUID Version-4

**Version-4 UUIDs are randomly generated.** There are over  $5.3 \times 10^{36}$  unique v4 UUIDs. This is the most common UUID version.

Version-4, variant-1 UUIDs are used in the game Minecraft. You can use our tool to [generate Minecraft UUIDs](#) than can be used in-game.

Version-4, variant-2 is called a "GUID" on Microsoft systems.

## Timestamp-first UUIDs

Timestamp-first are not mentioned in the UUID RFC; however, they are a common variation of version-4 UUIDs. This format is sometimes called "Ordered UUIDs" or "COMB" (combined time-GUID).

This version starts with the current time followed by randomness. There are two main reason for beginning UUIDs with the current timestamp:

1. When sorting by UUID they will appear in the order created
2. Ordered UUIDs can be more efficiently stored in indexed databases columns compared to random UUIDs

There are several variations of timestamp-first UUIDs in different implementations because there is not agreed upon specification. However, generally the first 8 hex digits represent the time and the remaining digits are random.

If you are using UUIDs as a primary key in a database you should use timestamp-first UUIDs.

## Nil UUIDs

This special case UUID is guaranteed to not be unique. **The Nil UUID is all zeros:**  
**00000000-0000-0000-0000-000000000000**



This UUID might be used as a template for the format of UUIDs or when you need an easily identified UUID that is guaranteed to not be unique.

**To determine the version of a UUID and to extract any information such as creation time (when available) use our [UUID decode tool](#).**

Free Online UUID/GUID Generator Tool.

[↑ Back to top](#)

[UUID Generator](#) • [Bulk UUID Generator](#) • [UUID Decoder](#) • [API Docs](#) • [API Status](#) • [Terms of service](#) • [Sitemap](#) • [Contact us](#)

 [UUIDTools.com](#) © 2021. All rights reserved.  [View on Github](#)