

**NAME**

**uuidgen** - generate universally unique identifiers

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <sys/uuid.h>
```

*int*

```
uuidgen(struct uuid *store, int count);
```

**DESCRIPTION**

The **uuidgen()** system call generates *count* universally unique identifiers (UUIDs) and writes them to the buffer pointed to by *store*. The identifiers are generated according to the syntax and semantics of the DCE version 1 variant of universally unique identifiers. See below for a more in-depth description of the identifiers. When no IEEE 802 address is available for the node field, a random multicast address is generated for each invocation of the system call. According to the algorithm of generating time-based UUIDs, this will also force a new random clock sequence, thereby increasing the likelihood for the identifier to be unique.

When multiple identifiers are to be generated, the **uuidgen()** system call will generate a set of identifiers that is dense in such a way that there is no identifier that is larger than the smallest identifier in the set and smaller than the largest identifier in the set and that is not already in the set.

Universally unique identifiers, also known as globally unique identifiers (GUIDs), have a binary representation of 128-bits. The grouping and meaning of these bits is described by the following structure and its description of the fields that follow it:

```
struct uuid {
    uint32_t  time_low;
    uint16_t  time_mid;
    uint16_t  time_hi_and_version;
    uint8_t   clock_seq_hi_and_reserved;
    uint8_t   clock_seq_low;
    uint8_t   node[_UUID_NODE_LEN];
};
```

*time\_low*                    The least significant 32 bits of a 60-bit timestamp. This field is stored in the native byte-order.

<i>time_mid</i>	The least significant 16 bits of the most significant 28 bits of the 60-bit timestamp. This field is stored in the native byte-order.
<i>time_hi_and_version</i>	The most significant 12 bits of the 60-bit timestamp multiplexed with a 4-bit version number. The version number is stored in the most significant 4 bits of the 16-bit field. This field is stored in the native byte-order.
<i>clock_seq_hi_and_reserved</i>	The most significant 6 bits of a 14-bit sequence number multiplexed with a 2-bit variant value. Note that the width of the variant value is determined by the variant itself. Identifiers generated by the <b>uuidgen()</b> system call have variant value 10b. the variant value is stored in the most significant bits of the field.
<i>clock_seq_low</i>	The least significant 8 bits of a 14-bit sequence number.
<i>node</i>	The 6-byte IEEE 802 (MAC) address of one of the interfaces of the node. If no such interface exists, a random multi-cast address is used instead.

The binary representation is sensitive to byte ordering. Any multi-byte field is to be stored in the local or native byte-order and identifiers must be converted when transmitted to hosts that do not agree on the byte-order. The specification does not however document what this means in concrete terms and is otherwise beyond the scope of this system call.

## RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

The **uuidgen()** system call can fail with:

[EFAULT]	The buffer pointed to by <i>store</i> could not be written to for any or all identifiers.
[EINVAL]	The <i>count</i> argument is less than 1 or larger than the hard upper limit of 2048.

## SEE ALSO

uuidgen(1), uuid(3)

## STANDARDS

The identifiers are represented and generated in conformance with the DCE 1.1 RPC specification. The **uuidgen()** system call is itself not part of the specification.

**HISTORY**

The **uuidgen()** system call first appeared in FreeBSD 5.0.