**What is GUID in c#?**

GUID (or UUID) is an acronym for 'Globally Unique Identifier' (or 'Universally Unique Identifier'). A GUID is a 128-bit integer (16 bytes) that can be used across all computers and networks wherever a unique identifier is required. Such an identifier has a very low probability of being duplicated.

**Why we use GUID?**

* We use GUID to create unique ID. A GUID makes a great primary key in a back-end database. While each generated GUID is not guaranteed to be unique, the total number of unique keys (2128 or 3.40282366×1038) is so large that the probability of the same number being generated twice is very small. For an application using 10 billion random GUIDs, the probability of a coincidence is approximately 1 in a [quintillion](http://en.wikipedia.org/wiki/Quintillion).

We may store users or products in your database and you want somehow uniquely identify each row in the database. Common approach is to create an auto incrementing integer; another way would be to create a GUID for your products.

**Where GUID Are used?**

* GUID are used in enterprise software development in C#, java and C++ as database keys or just about anywhere else, a truly unique identifier is required. GUID are also used to identify all interface and object in COM programming.

**Advantage of GUID**

* They really help in preventing data collisions.
* As GUID help in preventing data collisions it help with merging data between networks, machines, etc.
* It allows you to know your primary key before record is inserted.
* It cannot be easily guessed.

**Disadvantage of GUID**

* Each GUID occupies 16 bytes. Consider situation where database is storing 10000 entry then only GUID will occupies 160000 bytes.
* Most of PK are FK in other tables. Only 5 entry of FK consume 60 bytes to store only GUID. Over time, it become primary cause for performance issues.
* Random GUID is generated using algorithm that is out of order. For sorting, we have to use different algorithms as insertion timestamp.
* GUID are not much useful for small database.

**What is format of GUID?**

* GUID is a Structure, available in System Namespace
* GUID are most commonly written in text as a sequence of [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal) digits as such:
  + 3F2504E0-4F89-11D3-9A0C-0305E82C3301
  + Often braces are added to enclose the above format, as such:
  + {3F2504E0-4F89-11D3-9A0C-0305E82C3301}

**How to create GUID in c#?**

1. **Directly From visual Studio**

Open Visual Studio->Tools->Create GUID->Registry Format->**New GUID**

It will create new GUID every time you click **New GUID**.

1. **In console Application**

* The GUID method System.Guid.NewGuid() initializes a new instance of the GUID class.

using System;

namespace GUIDTest

{

class MainClass

{

static void Main(string[] args)

{

System.Guid guid=System.Guid.NewGuid();

Console.WriteLine(guid.ToString());

Console.ReadLine();

}

}

}

**Types of GUID**

1. **Date-time and timestamp (Version 1)**

* Version 1 use 48 bits MAC address with 60 bits of timestamp. In version 1 and 2 we use MAC address of network card. Usage of MAC address for node id means we can track computer that created version 1 GUID.
* We can track where document were created or edited through GUID (UUIDs) embedded into them by word processing software.
* We can replace MAC address with random 48 bits node id if node does not have MAC address or it is not desirable to expose it.

1. **DCE security (Version 2)**

* Version 2 GUIDs are similar to version 1, except that 8 bits of clock sequence are replaced by a “local domain” and 32 bits of first timestamp is replaced by integer, which is generated according to specified local domain.

1. **MD5 hash & namespace (Version 3)**

* Version 3 GUID are generated using hashing, namespace identifier and namespace name.
* Version 3 use MD5 as hashing algorithm.

1. **Random (Version 4)**

* Version 4 GUIDs are generated randomly, of the 128 bits in a GUID, 6 are reserved for special use (version + variant bits) giving us 122 bits that can be filled at random.
* The specification doesn't specify how the random numbers should be generated, they could be anything from psuedo-random to cryptographically secure - hence these GUIDs like all other GUIDs should only be used for identification and not for security.

**5. SHA-1 hash and message (Version 5)**

* Version 5 GUID are similar to version 3, except it use SHA-1 algorithm for hashing instead of MD5
* If you do not have to maintain backwards compatibility with existing MD5 GUIDs then SHA-1 is preferred.

**Structure of GUID**

* 3F2504E0-4F89-11D3-9A0C-0305E82C3301
* 4byte-2byte-2byte-2byte-6byte
* This text notation follows data structure as defined below. The sequence is
* Data1 (8 character)

time\_low: It is low 32 bits of timestamp

* Data2 (4 character)

time\_mid: It is middle 16 bits of timestamp

* Data3 (4 character)

time\_hi\_and\_version: 4 bit (1st character) represent GUID version and remaining are high 12 bits (3 character) of timestamp.

* Initial item of Data4 (4 character)

Clock\_seq\_hi and res\_clock\_seq\_low: 1st character represent variant of GUID and remaining 3 digit are high and low field of clock sequence.

* Remaining item of Data4 (12 character)

Node: 48 bit node id.

**Datatype of GUID(In SQL Server)**

* For specifying GUID datatype in SQL server, we have two option.

1. Uniqueidentifier

* There is some speed optimization when we use uniqueidentifier instead of varchar.

1. Varchar(36)

**Function and property of GUID**

1. **AutoSplitKey**

* A NULL GUID is valid but is not useful in a table. Therefore, the **AutoSplitKey** property is implemented for the GUID data type when it is used in a page. When GUID is selected as a primary key, **AutoSplitKey** is enabled for the page, and the GUID value remains NULL. When you create a new record, a valid GUID is created and assigned automatically.

**2. ISNULLGUID**

* ISNULLGUID takes a GUID value as a required argument and returns TRUE/FALSE depending on whether the GUID value is NULL. This function does not accept a Text value as an argument.
* Ex. Check= ISNULLCGUID(guid);

**3. CLEAR**

* CLEAR takes one required argument. It nullifies the GUID value.
* Ex. CLEAR(guid);

Cheers to coding!!!