

GUID چیست؟ بررسی مفهوم Globally Unique Identifier اکتیو دایرکتوری

زمانی که یک domain user و یا group account جدید ایجاد میشود سرویس Active Directory یک account's SID در Object-SID مربوط به user یا group ایجاد میکند، علاوه بر SID یک شناسه جدید منحصر بفرد در سطح جهان با نام GUID یا -Globally Unique Identifier- نیز به آن Object اختصاص میدهد GUID. یک مقدار ۱۲۸ بیتی است.

بر خلاف SID که یکتا بودن آن در سطح یک دامین تضمین شده است، یک GUID می بایست در سطح دنیا یکتا باشد. به عنوان مثال فرض کنید که یک کاربر در سطح دامین خود ایجاد نموده اید و به تبع آن، سرویس Active Directory برای آن یک GUID در نظر گرفته است. در این حالت، هر کجای دنیا که از سرویس Active Directory استفاده کنند، بنا به تعریف، نمی توانید کاربری با GUID یکسان با کاربر مورد نظر خود پیدا کنید. همچنین، بر خلاف SID که تنها برای کاربران، گروه ها، کامپیوتر اکانت ها و نهایتا سرویس ها در نظر گرفته می شود، هر object قابل تعریف در سرویس Active Directory دارای یک GUID یکتا می باشد. به منظور مشاهده GUID مرتبط با یک Object، طبق مسیر زیر عمل کنید:

```
Run->dsa.msc->view->advanced features-> right click on user->
properties-> attribute editor->object GUID
```

سرویس Active Directory از GUID ها به منظور انجام فعالیت های درونی خود استفاده می نماید GUID. از جمله مواردیست که از سوی Active Directory در Global Catalog ذخیره می گردد. بدین ترتیب می توان از امکانات جستجوی Global Catalog در خصوص یافتن object مورد نظر خود بر اساس GUID، در سطح کل forest استفاده نمود. در حقیقت، جستجوی یک object بر اساس GUID مرتبط با آن، مطمئن ترین روش جهت یافتن نتایج دلخواه می باشد. ارزش مقادیر دیگر attribute های مرتبط با یک object می تواند در سناریوهای مختلف تغییر نماید و این در حالیست که GUID مرتبط با یک object به هیچ عنوان تغییر نمی کند.

اما، SID می تواند در شرایطی تغییر کند. به عنوان مثال، هنگامی که یک کاربر از یک دامین به دامین دیگری منتقل می گردد، SID مرتبط با آن نیز بر اساس دامین مقصد، تغییر می کند. با توجه به این واقعیت و این موضوع که سطوح دسترسی تعریف شده بر روی منابع موجود در شبکه بر اساس SID مربوط به کاربران، گروه ها و یا کامپیوتر اکانت ها می باشد، این تغییر می تواند مانع از دسترسی کاربر مورد نظر گردد. لذا در این خصوص شاهد ارائه قابلیت از سوی

مایکروسافت در Windows Server ۲۰۰۳ هستیم. این قابلیت، SID History نام دارد. با استفاده از این قابلیت جدید، هنگامی که SID مربوط به یک object، به هر دلیلی تغییر میکند، SID قبلی آن از بین نرفته و در محل دیگری (SID History) ذخیره می گردد. توجه داشته باشید که این محل می تواند چندین SID را در خود جای دهد. به منظور مشاهده SID History مربوط به یک کاربر، می بایست طبق زیر عمل کنید:

```
Run->dsa.msc->view->advanced features-> right click on user->
properties->attribute editor -> sid History
```

هنگامی که یک کاربر با موفقیت عملیات احراز هویت را پشت سر می گذارد، یک Access Token با استفاده از SID فعلی کاربر، بعلاوه SID های قبلی آن (البته در صورت وجود) ایجاد گردیده و برای وی ارسال می گردد. حال اگر چنانچه کاربرد مورد نظر بخواهد به یک منبع دسترسی پیدا کند، وجود SID های مرتبط با کاربر می تواند در نحوه دسترسی به منبع مورد نظر موثر باشد. اما دلیل استفاده مایکروسافت از SID، در حالی که GUID موجود است، آنست که در زمان Windows NT، این سیستم عامل به منظور شناسایی کاربران و یا گروه ها، از SID استفاده مینمود. این موضوع با وجود GUID، همچنان بر قوت خود در مدرن ترین سیستم عامل های کنونی نیز باقیست.

چرا از GUID استفاده می کنیم؟

ما از GUID برای ایجاد یک شناسه منحصر به فرد استفاده می کنیم. یک GUID یک کلید اصلی عالی در پایگاه داده back-end می سازد. در حالی که هر GUID تولید شده تضمین نمی شود که منحصر به فرد باشد، تعداد کل کلیدهای یکتا (۱۰۳۸×۳.۴۰۲۸۲۳۶۶) آنقدر زیاد است که احتمال دوبار تولید همان عدد بسیار کم است. برای برنامه ای که از ۱۰ میلیارد GUID تصادفی استفاده می کند، احتمال تصادف تقریباً ۱ در کوئنتیلیون است (۱۰^{۳۰}).

ما ممکن است کاربران یا محصولات را در پایگاه داده خود ذخیره کنیم و بخواهیم به نحوی هر ردیف را در پایگاه داده به طور منحصر به فرد شناسایی کنیم. یک رویکرد رایج ایجاد یک عدد صحیح افزایش دهنده خودکار است. راه دیگر ایجاد یک GUID برای محصولات شما خواهد بود.

GUID ها کجا استفاده می شوند؟

یک سیستم توزیع شده را در نظر بگیرید که در آن بسیاری از برنامه نویسان به طور مستقل کار می کنند و شناسه تولید می کنند، در این سناریو این امکان وجود دارد که دو برنامه نویس شناسه مشترک تولید کنند و در زمان ادغام آن مشکل ایجاد کند، بنابراین زمانی که چندین سیستم مستقل یا کاربر داریم باید از GUID استفاده کنیم. ایجاد شناسه که باید منحصر به فرد باشد. در برنامه نویسی هر زمان که یک رابط یا متد ایجاد می کنیم یک شناسه منحصر

به فرد به آن داده می شود که برای فراخوانی آنها استفاده می شود. این شناسه منحصر به فرد GUID است. تولید GUID توسط برنامه ویندوز می تواند برای شناسایی فایل، برنامه، مؤلفه، ورودی پایگاه داده خاص استفاده شود.

مزیت GUID

- آنها واقعاً به جلوگیری از برخورد داده ها کمک می کنند.
- ادغام آسان پایگاه داده بین ماشین ها، شبکه های مختلف و غیره زیرا از برخورد داده ها جلوگیری می کند.
- این به شما امکان می دهد قبل از درج رکورد کلید اصلی خود را بدانید.
- به راحتی نمی توان حدس زد.

معایب GUID

- هر ۱۶ GUID بایت را اشغال می کند. وضعیتی را در نظر بگیرید که در آن پایگاه داده ۱۰۰۰۰ ورودی را ذخیره می کند. در آنجا فقط ۱۶۰۰۰۰ GUID بایت را اشغال می کند.
- بیشتر کلیدهای اصلی، کلیدهای خارجی در جداول دیگر هستند. فقط ۵ ورودی FK حدود ۶۰ بایت مصرف می کند تا فقط GUID را ذخیره کند. با گذشت زمان، علت اصلی مشکلات عملکرد می شود.
- GUID تصادفی با استفاده از الگوریتمی تولید می شود که از نظم خارج شده است. برای مرتب سازی، باید از الگوریتم های مختلف به عنوان مهر زمانی درج استفاده کنیم.
- GUID ها برای پایگاه های داده کوچک چندان مفید نیستند.

فرمت GUID چیست؟

- GUID ساختاری است که در فضای نام سیستم موجود است
- GUID ها معمولاً در متن به عنوان دنباله ای از ارقام هگزادسیمال نوشته می شوند.

○ ۳F۲۵۰۴E۰-۴F۸۹-۱۱D۳-۹A۰C-۰۳۰۵E۸۲C۳۳۰۱

○ اغلب بریس ها برای محصور کردن قالب فوق اضافه می شوند، به این ترتیب:

○ {۳F۲۵۰۴E۰-۴F۸۹-۱۱D۳-۹A۰C-۰۳۰۵E۸۲C۳۳۰۱}

چگونه GUID در سی شارپ ایجاد کنیم؟

۱. مستقیماً از ویژوال استودیو

Visual Studio->Tools->Create GUID->Registry Format->New GUID را باز کنید.

هر بار که روی **New GUID** کلیک کنید یک **GUID** جدید ایجاد می کند.

۱. **در Console Application**

متد **GUID System.Guid.NewGuid()** نمونه جدیدی از کلاس **GUID** را مقداردهی اولیه می کند.

```
using System;
namespace GUIDTest {
    class MainClass {
        static void Main(string[] args) {
            System.Guid guid = System.Guid.NewGuid();
            Console.WriteLine(guid.ToString());
            Console.ReadLine();
        }
    }
}
```

}