How to create a GUID

> ***Question 1***
>
> I'm looking for a function to create a GUID. I don't want to create a COM object or something, I just need a sort of 'Session ID' for a special database procedure.

```
uses
  ActiveX;

function GenerateKey: String;
var
  MyGUID: TGUID;
  MyWideChar: array[0..100] of WideChar;
begin
  {First, generate the GUID:}
  CoCreateGUID(MyGUID);
  {Now convert it to a wide-character string:}
  StringFromGUID2(MyGUID, MyWideChar, 39);
  {Now convert it to a Delphi string:}
  Result := WideCharToString(MyWideChar);
  {Get rid of the three dashes that StringFromGUID2() puts in the result
  string:}
  while Pos( '-', Result ) > 0 do
    Delete( Result, Pos( '-', Result ), 1 );
  {Get rid of the left and right brackets in the string:}
  while Pos( '{', Result ) > 0 do
    Delete( Result, Pos( '{', Result ), 1 );
  while Pos( '}', Result ) > 0 do
    Delete( Result, Pos( '}', Result ), 1 );
end;
```

Tip author unknown

> ***Question 2***
>
> I have a very simple piece of test code to generate and display a GUID on the screen every time a user clicks on a button on a form. When I compile the program and run the executable on my Windows 2000 machine, I get a unique GUID every time I click the button, which is what I expect from the documentation. However, when I run the same executable on any Windows 98 SE machine, and even a Windows NT 4.0 Server (with SP5), it simply generates the exact same GUID over, and over, and over again.

Using RAW API:

```
{ ... }

var
  Form1: TForm1;
  UuidCreateFunc : function (var guid: TGUID):HResult;stdcall;

implementation

{$R *.DFM}

uses
  ComObj;

procedure TForm1.Button1Click(Sender: TObject);
var
  hr: HRESULT;
  m_TGUID: TGUID;
  handle: THandle;
begin
  handle := LoadLibrary('RPCRT4.DLL');
  @UuidCreateFunc := GetProcAddress(Handle, 'UuidCreate') ;
  hr := UuidCreateFunc(m_TGUID);
  if failed(hr) then
```

```
    RaiseLastWin32Error;
  ShowMessage(GUIDToString(m_TGUID));
end;
```

With WIN2K support:

```
{ ... }

var
  Form1: TForm1;
  UuidCreateFunc: function (var guid: TGUID): HResult; stdcall;

implementation

{$R *.DFM}

uses
  ComObj;

procedure TForm1.Button1Click(Sender: TObject);
var
  hr: HRESULT;
  m_TGUID: TGUID;
  handle: THandle;
  WinVer: _OSVersionInfoA;
begin
  handle := LoadLibrary('RPCRT4.DLL');
  WinVer.dwOSVersionInfoSize := sizeof(WinVer);
  getversionex(WinVer);
  if WinVer.dwMajorVersion >= 5 then  {Windows 2000 }
    @UuidCreateFunc := GetProcAddress(Handle, 'UuidCreateSequential')
  else
    @UuidCreateFunc := GetProcAddress(Handle, 'UuidCreate') ;
  hr := UuidCreateFunc(m_TGUID);
  if failed(hr) then
    RaiseLastWin32Error;
  ShowMessage(GUIDToString(m_TGUID));
end;
```

Tip by Ivan Kozhuharov

Original resource:	The Delphi Pool
Author:	Anon & Ivan Kozhuharov
Added:	2013-03-13
Last updated:	2013-03-13