

## DSS: ZÁPISY UNIKÁTNÍ IDENTIFIKACE POMOCÍ GUID

Unikátní GUID (Global Unique Identifier) je cestou, jak se od přepisování posunout ke strojovému zpracování záznamu. Identifikuje totiž záznamy v DSS.

Stejně jako jakékoliv jiné vyspělé datové standardy určené pro strojové zpracování pomocí různých softwarových nástrojů, využívá také Datový standard staveb (DSS) pro unikátní identifikaci záznamů v rozsáhlejších databázích tzv. GUID (Global Unique Identifier), který bývá někdy označován také jako UUID (Universal Unique Identifier). Využití tohoto standardu při tvorbě DSS je v souladu s požadavky ČSN EN 23386 pro tvorbu datových slovníků. Tím také technicky DSS je.

Využití má GUID zejména pro softwarové vývojáře, protože právě v jejich nástrojích má svou roli. Pro odbornou stavařskou veřejnost tedy nemá prakticky žádný význam. Pro skutečnou digitalizaci se totiž musíme posunout od „ručního“ přepisování informací, či používání excelových tabulek, ke strojovému zpracování různými softwarovými nástroji či jejich doplňky. Protože teprve potom budeme těžit ze zvýšení produktivity a snížení chybovosti, které si od digitalizace slibujeme. Spolehlivé, stabilní a jednoznačné označování záznamů (u DSS např. datová šablona, vlastnost, účel užití, milník) je zásadním prvkem, bez kterého by rozvoj softwarových nástrojů nebyl možný. Nezastupitelnou roli hraje GUID jak při identifikaci aktualizací obsahu záznamů, tak – a možná především – při návrhu dalších pokročilých funkcionalit. Může jít například o scénáře kontrol nebo třeba různé průvodce, kteří přinesou uživatelům snadnější zadávání a vytváření **digitálních modelů staveb (DiMS)**.

### Technický popis pro programátory

Na následujících řádcích uvádíme technický popis určený zejména pro programátory. Ti jej mohou využít například pro konverzi zápisu GUIDů mezi jejich různými druhy zápisů.

- GUIDy jako takové reprezentují 128-bitové číslo, které má několik možných reprezentací v různých číselných soustavách.
- Různé kódování zápisu pomocí různých číselných soustav je především motivováno úsporou místa pro zápis v rozsáhlých databázích a tedy k jejich zmenšování.
- Níže uvádíme základní zápis pomocí 16-kové soustavy a pak dva příklady, které využívá organizace buildingSmart (64-ková a 85-ková soustava).
- Nejčastěji je použita reprezentace GUID v publikovaných datech pomocí 16-kové soustavy s využitím cifer '0123456789abcdef', a rozdělené oddělovačem v podobě pomlčky po skupinách 8-4-4-4-12 cifer.
- Jiná reprezentace, používaná například organizací buildingSmart, využívá BASE64 kódování v 64-kové soustavě s využitím cifer  
'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz\_&#39;
- Podle dalšího **standardu BuildingSMART** existuje i kódování pomocí osmdesáti pětkové soustavy s využitím cifer  
'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!#\$%&^|\*+,-./:;<=>?~`@\_&#39;

Na stránce <https://technical.buildingsmart.org/resources/ifcimplementationguidance/ifc-guid/> jsou odkazované i zdrojové texty pro konverzi do tohoto formátu v C# a Pythonu.

## PŘÍKLADY FUNKCÍ PRO PŘEVOD, JAK JE POUŽÍVÁME MY V NAŠICH AKTUÁLNÍCH NÁSTROJÍCH PRO PŘÍPRAVU DSS:

Pro převod mezi formáty používáme dvě funkce v PL/SQL, které reprezentaci převedou nejprve z původní soustavy na klasické číslo, a následně z něj do požadované soustavy.

```
function UUID_16TO64(X varchar2) return varchar2
```

```
as
```

```
Conv64 CONSTANT VARCHAR2(64) :=  
'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_$';  
--  
N NUMBER;  
K NUMBER;  
I NUMBER;  
G VARCHAR2(32);  
begin  
-- 00000000-0000-0000-0000-000000000000  
I := 0;  
for K in 1 .. nvl(length(X),0) loop  
if substr(X,K,1)<>'-'  
then  
I := 16*I + instr(Conv64,upper(substr(X,K,1)))-1;  
end if;  
end loop;  
-- cislo  
N := I;  
G := '';  
for I in 1 .. 22 loop -- ceil(128/6)  
K := mod(N,64);  
N := TRUNC(N/64);  
G := substr(Conv64,K+1,1) || G;  
end loop;  
return G;  
end UUID_16TO64;
```

```
create or replace function UUID_64TO16(X varchar2) return varchar2
```

```
as
```

```
Conv64 CONSTANT VARCHAR2(64) :=  
'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_$';  
--  
N NUMBER;  
K NUMBER;  
I NUMBER;  
G VARCHAR2(32);  
--  
begin  
if X is null  
then  
return '';  
end if;
```

```
--  
N := 0;  
for K in 1 .. nvl(length(X),0) loop  
  I := INSTR(Conv64,SUBSTR(X,K,1));  
  IF I>0  
  THEN  
    N := 64*N + (I-1);  
  END IF;  
end loop;  
--  
G := "";  
for I in 1 .. 32 loop  
  K := mod(N,16);  
  N := TRUNC(N/16);  
  G := lower(substr(Conv64,K+1,1)) || G;  
end loop;  
-- 00000000-0000-0000-0000-000000000000  
return substr(G,1,8) || '-' || substr(G,9,4) || '-' || substr(G,13,4) || '-' || substr(G,17,4) || '-' || substr(G,21,12);  
end UUID_64TO16
```