



# PUID



Property **U**nique **I**Dentifier

Paul Desormeaux – EquiNet Inc.



## Why? - Business Cases

- **Deduplication** - DB (MLS, Vendor, Websites, etc.)
- **Aggregation** – Multiple sources (mash-up: Green data)
- **Fraud Prevention** – Uniqueness of mortgage contracts
- **Prop History Report / CDOM** – Even with address ‘typo’
- **Internet Tracking** Value to MLS, Vendor, Broker, Agent, ...





## Situation

- Multiple home-grown '**dedup**' solutions exist in the wild
- These silo solutions work internally only, with varying success

## Goal

- Raise the bar via standardization
- Avoid duplicate development & maintenance efforts
- Ultimately move the PUID outside the silos allowing property information to flow and effectively mash-up



## Perfect World

- (**Latitude, Longitude, Elevation**) **DD**: 8 decimal places = **1 mm**
- Stick tricorder on main point of egress or middle of lot polygon

## Real (Estate) World

- Use property **data at hand** (MLS, whatever)
- What data (**DD** fields) to use? See next slide!
- Design **PUID** to be ready to adapt when perfect world happens!



# What is a PUID?



- *Listing* attribute identifying real property:
  - Usually: House, Lot, Condo, Farm, Warehouse, etc.
  - Can be: For Rent, Timeshare, Bus. Opp., Mobile Home, etc.
- Usually has Tax ID info –but not necessarily!

## What if No Tax ID Info?



- Use Address info on hand, if available?
- Use Latitude, Longitude info on hand, if available?
- Eureka!: Don't actually need a PUID # proper! Just need a way to compare certain listing fields to determine the likelihood two listings are associated with the same property.



# ComparePUID()

- **PUID** not a classical ID with N alpha-numeric chars
- **PUID** is made up of a **subset of DD fields** which can be used to determine uniqueness when intelligently compared

## What Kind of DD Fields?

- **FOR**: Tax fields (i.e. **Country + State + County + Parcel#**) 
- **ADR**: Address fields (i.e. **StreetNumber, StreetName, Country**)
- **GEO**: Geographic fields (i.e. **Latitude, Longitude, Elevation**)



# PUID = DD Field Subset

**PUID** =



**DD** Fields

Think new PUID field made-up of (FOR, ADR, GEO) DD fields.

Or just a DB *View*.



## Expect / Guess

- **FOR**: 95%
- **ADR**: 90%
- **GEO**: 85%



99% ???

FOR	ADR	GEO
X		
	X	
		X
X	X	
	X	X
X		X
X	X	X





# FOR



Field Name
Country
CountyOrParish
ParcelNumber
StateOrProvince
TaxBlock
TaxBookNumber
TaxLot
TaxMapNumber
TaxParcelLetter
TaxTract
Township
UnitNumber
PublicSurveySection
PublicSurveyTownship
PublicSurveyRange



# ADR



Field Name
City
Country
CountyOrParish
PostalCity
PostalCode
PostalCodePlus4
StateOrProvince
StreetDirPrefix
StreetDirSuffix
StreetSuffixModifier
StreetName
StreetNumber
StreetNumberNumeric
StreetSuffix
CrossStreet



# GEO



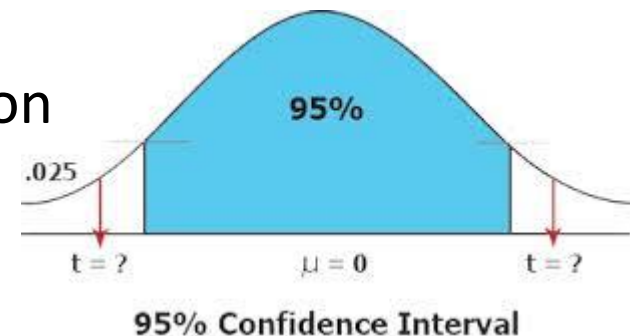
Field Name
Elevation
EntryLevel
ElevationUnits
Latitude
Longitude
FrontageType
WaterfrontYN
WaterBodyName
<b>PropertyType</b>
<b>PropertySubType</b>
<b>LotSizeArea</b>
<b>LotSizeUnits</b>
<b>LotSizeAcres</b>
<b>LotSizeSquareFeet</b>
<b>ListAOR</b>





## Details

- Detailed field **analysis** performed on sample property listings provided by various MLSs (**Availability, Quality, Dependability**)
  - Required **data cleansing & normalization**
  - How to compare each field: individually or collectively
  - **Compare**: Equality, *Fuzzy*, # decimal places, etc.
- Output of **ComparePUID()** will be a **confidence level** of properties being the same (**0.0 – 1.0**)
- **Versioned**: request version, used version



## Now / Next ...

- Starting to write & reference algorithms & **pseudo code**
- More info needed:
  - DD **Tax\*** field usage
  - Field **stats** on larger property listing sample space
- Code and test via GitHub or RESO's LineNode.com
- Test internally until **V1** ready
- Publish and have parties to test internally, get feedback
- Setup an iterative process of releasing **ComparePUID()** versions via successive refinements based on feedback
- Determine **ComparePUID()** input parameters (i.e. **ListAOR**, **#DecimalPlacesLatLong**, etc.)





## FUTURE >>>

- Add to **DD** Resources? **Property, Open House**, etc.
- Use in **Internet Tracking**?
- Create a parametrized framework to easily specify, update and adjust **versioned cleansing, normalization** and **comparison** functions
- Create a RESO **ComparePUID()** Web Service (WS)
- Accumulate usage and information to feedback into refining the comparison algorithm
- **NOTE**: Still option to create **GeneratePUID()** to generate a usable **PUID** value (i.e. hash based on **FOR** fields only) –but this will will never be as powerful as **ComparePUID()**



# PUID QUESTIONS?



Paul Desormeaux – PUID Workgroup Chair – [lpd@equinet.ca](mailto:lpd@equinet.ca)

